

**Submitted in partial fulfillment of the requirements of the Degree
of Doctor of Philosophy**

**Modelling and Interactional Control of a
Multi-fingered Robotic Hand for Grasping
and Manipulation**

Md Rakibul Hasan

26/09/2014



Queen Mary, University of London
School of Engineering and Materials Science

I, Md Rakibul Hasan, confirm that the research included within this thesis is my own work or supported by others, that this is duly acknowledged and my contribution indicated.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis. I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

Signature:

Date: 26/09/2014.

Abstract

In this thesis, the synthesis of a grasping and manipulation controller of the Barrett hand, which is an archetypal example of a multi-fingered robotic hand, is investigated in some detail. This synthesis involves not only the dynamic modelling of the robotic hand but also the control of the joint and workspace dynamics as well as the interaction of the hand with object it is grasping and the environment it is operating in. Grasping and manipulation of an object by a robotic hand is always challenging due to the uncertainties, associated with non-linearities of the robot dynamics, unknown location and stiffness parameters of the objects which are not structured in any sense and unknown contact mechanics during the interaction of the hand's fingers and the object. To address these challenges, the fundamental task is to establish the mathematical model of the robot hand, model the body dynamics of the object and establish the contact mechanics between the hand and the object.

A Lagrangian based mathematical model of the Barrett hand is developed for controller implementation. A physical SimMechanics based model of the Barrett hand is also developed in MATLAB/Simulink environment. A computed torque controller and an adaptive sliding model controller are designed for the hand and their performance is assessed both in the joint space and in the workspace. Stability analysis of the controllers are carried out before developing the control laws. The higher order sliding model controllers are developed for the position control assuming that the uncertainties are in place. Also, this controllers enhance the performance by reducing chattering of the control torques applied to the robot hand.

A contact model is developed for the Barrett hand as its fingers grasp the object in the operating environment. The contact forces during the simulation of the interaction of the fingers with the object were monitored, for objects with different stiffness values. Position and force based impedance controllers are developed to optimise the contact force. To deal with the unknown stiffness of the environment, adaptation is implemented by identifying the impedance. An evolutionary algorithm is also used to estimate the desired impedance parameters of the dynamics of the coupled robot and compliant object.

A Newton-Euler based model is developed for the rigid object body. A grasp map and a hand Jacobian are defined for the Barrett hand grasping an object. A fixed contact model with friction is considered for the grasping and the manipulation control. The compliant dynamics

of Barrett hand and object is developed and the control problem is defined in terms of the contact force. An adaptive control framework is developed and implemented for different grasps and manipulation trajectories of the Barrett hand. The adaptive controller is developed in two stages: first, the unknown robot and object dynamics are estimated and second, the contact force is computed from the estimated dynamics. The stability of the controllers is ensured by applying Lyapunov's direct method.

Nomenclature

τ_i	Joint torque of each Barrett finger.
c_b	Reference/base frame of the Barrett hand.
$M_{fi}(q) \in \mathbb{R}^{n \times n}$	The inertia matrix of the robot dynamics.
$M_{oi}(x) \in \mathbb{R}^{n \times n}$	The inertia matrix of the object dynamics.
q_{di}	Desired joint angle (deg) of each Barrett finger.
$C_{fi}(q, \dot{q}) \in \mathbb{R}^{n \times 1}$	The Coriolis vector terms of the robot dynamics.
$C_{oi}(x, \dot{x}) \in \mathbb{R}^{n \times 1}$	The coriolis and gravity vector terms of the object dynamics.
$G_{fi}(q) \in \mathbb{R}^{n \times 1}$	The gravity vector terms of the robot dynamics.
s_i	Sine values of the joint q_i .
Y_i	Regressors matrix of the robotic hand.
c_i	Cosine values of the joint q_i .
F_i	Finger name of the Barrett hand.
l_i	Link length of the robot finger.
m_i	Mass of each link of the Barrett hand.
q_i	Revolute joint angle (deg) of each Barrett finger.
θ_i	Revolute Initial angle (deg) of each Barrett finger.
α_i	Spread angle (deg) of finger F_1 and F_2 .
x_i, y_i, z_i	Initial fingertip position (m) of the Barrett finger.
x_f, y_f, z_f	Final fingertip position (m) of the Barrett finger.

Contents

Abstract	3
Nomenclature	5
Acknowledgments	6
1. Introduction	7
1.1. Multi-fingered hand features	8
1.2. Grasping and manipulation problem	9
1.3. Recent developments	12
1.4. Motivations, aim and objectives	18
1.5. Contributions	19
1.6. Outline of the thesis	20
1.7. Publications	21
2. Modelling of the Barrett hand and the hand-object system	23
2.1. Introduction	23
2.2. The Barrett hand description	23
2.3. Hand Kinematics	25
2.3.1. Forward Kinematics	25
2.3.2. Inverse Kinematics	26
2.4. Barrett hand dynamics	29
2.4.1. Finger dynamics	29
2.4.2. Energy calculation for <i>Lagrangian</i>	30
2.4.3. Dynamics of the fingers F_1 and F_2	30
2.4.4. Dynamics of the finger F_3	31
2.4.5. Dynamics of the complete hand	32
2.4.6. Structural properties of hand dynamics	33
2.4.7. Ratio dynamics and torque switch	34
2.5. Workspace dynamics	35

2.6.	Physical model of the Barrett hand	36
2.6.1.	SimMechanics for physical modelling	37
2.6.2.	Conversion of Barrett CAD into SimMechanics	37
2.7.	Object Dynamics	40
2.8.	Summary	43
3.	Adaptive and Higher order sliding mode control of the Barrett hand	45
3.1.	Introduction	45
3.2.	Equilibrium and Lyapunov Stability	47
3.3.	Motion problem formulation	50
3.4.	Computed torque control	51
3.4.1.	Proof of computed torque control stability	52
3.4.2.	Computed torque control design for the Barrett hand	53
3.4.3.	Results and discussions	55
3.5.	Sliding and adaptive sliding mode control	65
3.5.1.	Sliding mode theory	65
3.5.2.	SMC Implementation on planar finger	67
3.5.3.	Simulation and results of sliding mode control	67
3.5.4.	Adaptive SMC theory (ASMC)	69
3.5.5.	Simulation and results of ASMC	70
3.5.6.	Barrett hand simulation with SMC and ASMC	73
3.6.	Higher order sliding mode	79
3.6.1.	Background	79
3.6.2.	Higher Order Sliding Mode structure	79
3.6.3.	Higher order sliding mode control laws	83
3.6.4.	Higher Order Sliding Mode Control with adaptation (HOSMA)	84
3.6.5.	HOSM and HOSMA control implementation to planar finger	85
3.6.6.	HOSMA control implementation to Barrett finger	87
3.7.	Results and discussions	87
3.7.1.	Comparison of the controllers	94
4.	Interactional kinematics, dynamics and force control of the Barrett hand	96
4.1.	Introduction	96
4.2.	Contact mechanics	97
4.3.	Problem formulation for force control	100
4.4.	Force control techniques	101
4.5.	Impedance control (IMC)	102
4.5.1.	IMC objective and its structure	103
4.5.2.	IMC classifications	105
4.5.3.	Contact force analysis and results	107

4.5.4.	Planar finger control with PIM	112
4.5.5.	Planar finger control with FIM	115
4.5.6.	Position and force impedance control of Barrett finger	117
4.6.	Adaptive force impedance control	119
4.6.1.	Adaptive force impedance method of planar finger	121
4.6.2.	Adaptive force impedance method of Barrett finger	122
4.7.	Evolutionary algorithm based impedance control	122
4.7.1.	Evolutionary algorithm	122
4.7.2.	Discretized impedance controller	126
4.7.3.	Simulation procedure	128
4.7.4.	Results and discussions	128
4.8.	Summary	131
5.	Classical and adaptive controller implementation for grasping and manipulation	133
5.1.	Introduction to grasping and manipulation	133
5.2.	Definition of Grasping	134
5.2.1.	Grasp planning	134
5.2.2.	Grasp properties	134
5.2.3.	Grasp map	137
5.2.4.	Force-Closure of a grasp	138
5.2.5.	Hand Jacobian	139
5.2.6.	Grasp constraint	140
5.3.	Robot Object dynamics	140
5.3.1.	Multi-fingered Grasping and manipulation problem formulation	142
5.4.	Computed torque controller	143
5.4.1.	Stability of the CTC law	144
5.4.2.	Computed torque control for Barrett hand grasping and manipulation	145
5.4.3.	CTC results and discussion	146
5.5.	Natural and stiffness controller	154
5.5.1.	Results and discussions	156
5.6.	SMC-adaptive controller for grasping and manipulation	160
5.6.1.	Hand-object dynamics for SMC-adaptive controller	161
5.6.2.	The SMC-adaptive controller implementation	161
5.6.3.	Desired external and contact force calculation	162
5.6.4.	Stability of the adaptive controller	165
5.7.	Simulation	166
5.8.	Results and discussions	166
5.8.1.	Comparison of the controllers	173

6. Conclusions and Future Work	175
6.0.2. Future directions	178
A. Appendix	179
A.1. Stability of the SMC controller	179
References	181

List of Figures

2.1.	The Barrett hand CAD model in its initial position.	24
2.2.	Side view of The Barrett hand CAD model.	25
2.3.	The Barrett hand forward kinematics.	26
2.4.	The workspace range of finger F_1 for joint angle q_i and α_1 calculation.	28
2.5.	The workspace range of finger F_3 for joint angle q_i calculation.	28
2.6.	The motion based conditional mechanism of <i>torqueswitch</i>	34
2.7.	Single rigid robot link with one DOF revolute joint.	37
2.8.	Single rigid robot link with one DOF revolute joint.	38
2.9.	SimMechanics model architecture of the Barrett hand.	39
2.10.	Physical SimMechanics model of the three fingered Barrett hand.	40
3.1.	CTC Simulink based controller for the Barrett hand simulation.	54
3.2.	The damping added to the outer joint for balancing outer link (Finger F_3).	54
3.3.	Actual fingertip position tracking of the Barrett fingers for a desired Cartesian trajectory.	56
3.4.	Joint actual (blue) position tracking of the Barrett fingers for the joint position $q_2 = 45^\circ$ and $q_3 = 55^\circ$	57
3.5.	Joint velocities (\dot{q}_i) of Barrett fingers for the joint position, $q_2 = 45^\circ$ and $q_3 = 55^\circ$	58
3.6.	Joint torque (τ) of the Barrett fingers for the joint position, $q_2 = 45^\circ$ and $q_3 = 55^\circ$	59
3.7.	Joint space tracking of Barrett fingers in Simulink for the spread position, $\alpha = 65^\circ$, joint position, $q_2 = 45^\circ$ and $q_3 = 55^\circ$	61
3.8.	Joint space tracking of the Barrett fingers in SimMechanics for the spread position, $\alpha = 65^\circ$, joint position, $q_2 = 45^\circ$ and $q_3 = 55^\circ$	62
3.9.	Comparison of the joint tracking error in SimMechanics and Simulink for the spread position, $\alpha = 65^\circ$, joint position, $q_2 = 25^\circ$ and $q_3 = 35^\circ$	63
3.10.	Comparison of the joint tracking error in SimMechanics and Simulink for the spread position, $\alpha = 45^\circ$, joint position, $q_2 = 25^\circ$ and $q_3 = 35^\circ$	64
3.11.	SMC based control structure applied to a planar finger.	67

3.12. Joint position, velocities, control torque and sliding function results with the SMC controller applied to a planar finger.	68
3.13. Adaptive-SMC based control structure applied to a planar finger.	70
3.14. Adaptive SMC results for joint motion q , velocity \dot{q} , torque τ and sliding function s	71
3.15. Parameter estimation of σ_i with adaptive update law.	72
3.16. Tracking errors obtained with SMC and ASMC for two link planar joints.	72
3.17. Joint tracking results of the Barrett Hand with SMC.	74
3.18. Joint tracking results of the Barrett Hand with ASMC.	75
3.19. Comparison of Barrett finger tracking errors obtained with SMC and ASMC controllers.	76
3.20. Parameter estimation of σ_i with adaptive update law of Barrett finger F_1 and F_2	77
3.21. Parameter estimation of σ_i with adaptive update law of Barrett finger F_3	77
3.22. Sliding function s_i of the Barrett hand.	78
3.23. Joint torque τ_i calculated with Adaptive-SMC law for the Barrett hand.	78
3.24. Super twisting algorithm phase trajectory	83
3.25. Control structure of HOSM based control for a planar finger.	85
3.26. Comparison of Joint position between HOSMA and other controllers	86
3.27. Planar joint tracking error obtained with HOSMA controller (with noise).	87
3.28. Comparison of control signal and sliding function between HOSMA and other controllers	88
3.29. Joint positions q_2 and q_3 of finger F_3 with controller A: HOSM	89
3.30. Joint positions q_2 and q_3 of finger F_3 with controller A: HOSMA	89
3.31. Joint errors q_2 and q_3 of finger F_3 with controller A : HOSM (left) and HOSMA (right).	90
3.32. Joint errors q_2 and q_3 of finger F_3 with controller B : HOSM (Left) and HOSMA (Right).	90
3.33. Joint errors q_2 and q_3 of finger F_3 with super twisted controller C : HOSM (Left) and HOSMA (Right).	91
3.34. Minimum and maximum tracking errors obtained from controllers A,B and C in HOSM and HOSMA mode.	92
3.35. Joint torques of finger F_3 with controller A : HOSM (Left) and HOSMA (Right).	92
3.36. Joint torques of finger F_3 with controler B : HOSM (Left) and HOSMA (Right).	93
4.1. Contact mechanics between F_3 and an object with linear.	98
4.2. (i) Spring mechanics (ii) Spring Damper mechanics.	103
4.3. Position based impedance control structure for manipulator.	106

4.4. Position, force and torque results of one link planar finger during the interaction with an object.	108
4.5. Contact force measurement model of Barrett hand.	109
4.6. Contact force F_c calculation of Barrett Finger F_3 for different object stiffness K_e values.	110
4.7. Reaction torque τ_d of the finger F_3 joints. Top left $K_s = 10^3 N/m$, Top right $K_s = 20 kN/m$, Bottom left $K_s = 30 kN/m$, Bottom right $K_s = 50 kN/m$	111
4.8. Fingertip motion of F_3 (Top) and joint velocities \dot{q}_i (Bottom).	111
4.9. Contact force and reaction torque measured in workspace based control of finger F_3	112
4.10. Position based impedance control schematics of planar finger.	112
4.11. SimMechanics contact model between robot link and object.	113
4.12. Position based impedance control results from planar finger.	114
4.13. Force based impedance control diagram for planar finger.	115
4.14. Force based impedance control results from planar finger.	116
4.15. Position Impedance control of Barrett finger F_3 , $K_e = 10^3 N/m$	118
4.16. Position based impedance control of the Barrett finger F_3 for different stiffness.	119
4.17. Force based impedance control of Barrett finger for desired force $F_d = 3N$, stiffness value, $k_e = 10^3 N/m$	120
4.18. Adaptive impedance control of planar fingers for desired force $F_d = 30N$	123
4.19. Adaptive impedance control of Barrett finger for desired force $F_d = 10N$	124
4.20. CMS-EA based impedance parameter selection process from digital impedance controller based robot-object interactional environment.	127
4.21. Impedance parameter estimation (Top); Fitness function, standard deviation computation (Bottom) with $M_{Io}, K_{Io} = [0.5, 0.5]$	129
4.22. Contact force optimisation with estimated impedance parameters through evolution.	129
4.23. Impedance parameter estimation (Top); Fitness function, standard deviation computation (Bottom) with $M_{Io}, K_{Io} = [5, 5]$	130
4.24. Contact force optimisation with estimated impedance parameters through evolution.	130
5.1. The Barrett hand grasping an object.	135
5.2. Point contact (Left), Point contact with friction (Middle) and Soft contact (Right).	136
5.3. Contact model based on Coulomb's friction.	137
5.4. Frame transformation between object and velocities in a multi-fingered grasp.	140
5.5. General control structure for grasping and Manipulation by the Barrett hand.	145
5.6. General control structure in the hand coordinates before grasping by the Barrett hand.	146

5.7. General control structure in the object space after grasping by the Barrett hand.	147
5.8. Control structure in the object space after grasping by the Barrett hand.	147
5.9. Tracking results of joint motions q_i in the Cartesian space before grasp state. . .	149
5.10. Joint velocities \dot{q}_i of a Barrett finger in the Cartesian space before the grasp state.	150
5.11. Joint torques τ_i of the Barrett fingers before grasping.	151
5.12. Normal contact forces F_n of Barrett fingers during grasp	152
5.13. Object motion X_o tracking after grasped by the Barrett hand.	153
5.14. CTC based Tracking error e_o of the object after grasped by the Barrett Hand. .	154
5.15. Joint torques τ_i of the Barrett fingers after grasping.	155
5.16. Position tracking of the object after grasped by the Barrett hand with natural controller.	157
5.17. Object position tracking after grasped by the Barrett hand with the stiffness controller (case 1).	158
5.18. Object position tracking after grasped by the Barrett hand with the stiffness controller (case 2).	158
5.19. Comparison of object motion based on CTC, Natural and Stiffness control al- gorithm.	159
5.20. General control structure in workspace before grasping by The Barrett Hand. .	164
5.21. Joint position responses of the Barrett fingers before and after grasp.	167
5.22. Linear motion of the object after grasp is made by the Barrett hand.	168
5.23. Linear velocity results of the object after grasp is made by the Barrett hand. . .	169
5.24. Orientation results of the object after grasp is made by the Barrett hand. . . .	169
5.25. Estimated mass parameters of the integrated dynamics with finger F_1	170
5.26. Estimated mass parameters of the integrated dynamics with finger F_3	170
5.27. Contact force errors after the object grasped by the hand.	171
5.28. Sliding function errors of Barrett fingers.	172
5.29. Angular velocities of the object after grasp.	172

List of Tables

2.1. Static joint positions of the Barrett hand and their motion range.	24
2.2. Workspace range of the Barrett hand.	29
3.1. Initial joint and workspace positions of the Barrett Hand.	55
3.2. Planar joint errors analysis with SMC and ASMC controllers.	73
3.3. Tracking error and torque analysis of Barrett finger F_3 with HOSM and HOSMA.	93
3.4. Comparison of the implemented controllers applied to the planar and the Bar- rett finger.	95
4.1. Impedance and admittance function for spring and spring damper model.	103
4.2. Impedance and admittance function for spring and spring damper model.	131
5.1. Fingertip positions of the Barrett hand at initial and contact state.	146
5.2. Initial $X_o(o)$ and desired $X_{do}(t)$ positions of the Object for grasping.	152
5.3. Range of settling time, tracking error and torque profiles of the joints and the object (before and after grasp).	154
5.4. Range of settling time, tracking error and torque profiles of the joints and the object (before and after grasp).	157
5.5. Position tracking errors obtained from Figure 5.22.	168
5.6. Orientation tracking errors obtained from Figure 5.24.	169
5.7. Result analysis of object motion, contact forces, parameter estimation and slid- ing errors.	173
5.8. Comparison of the implemented controllers for grasping and manipulation by the Barrett hand.	174

Acknowledgments

I would like to specially thank my first supervisor Dr Mohammad Hasan Shaheed for his valuable guidance, patience, enthusiasm and critical comments. He provided an exciting, stimulating and open research environment at the department, which encouraged me to pursue my research eagerly. I would also like to thank my second supervisor Dr Henri Huijberts for his support. A special thanks to Dr Ranjan Vepa for his constructive advise on advanced control applications and overall guidance. It has been a great honour for me to learn from them. I must also acknowledge the unconditional love and support of my parents. I cannot express with words all my gratitude to my parents for being the real reason that I am here now and doing what I do. They worked hard to provide me with an education that would ensure me a life that they did not have. Without their support, it would not be possible to start, continue and accomplish my study abroad.

Md Rakibul Hasan

1. Introduction

From the beginning of the ancient age, human hands have made an enormous contribution to the progress and achievements of human civilization. Current advanced civilisation is the reflection of human thoughts accomplished by human hands. The superiority of the human hand and its evolution with thought is one of the most important reasons for the success of humans in comparison to our closely related primates. Human hands possess a level of skill that always has advantages over the other parts of the bodies for executing daily tasks (picking, manipulating and placing of the objects) efficiently. Hence, the human always looked for alternatives to replace their hands for faster execution of these tasks by saving time and energy. This approach helped to build automated or controlled mechanisms for performing activities similar to a human being and led to the development of biomimetic robotics. In general, biomimetic robotic manipulators are designed based on human limbs. A robotic manipulator with a robot hand mimicking the human hand is exceedingly effective in all its current and future applications. The manipulator is composed of two parts: the joint based mechanical arm and the end effector. The mechanical arm plays the role to reach the target object and the end-effector is responsible for holding the object and manipulate it.

Robotic hands require more skills and dexterity to accomplish tasks. For example, picking a rough shaped object is carried out in two steps: first, reach the object location by the manipulator arm, and then the hand requires enough flexibility to grasp the object considering its complex shape. This requirement made the necessity to develop multi-fingered robot hand with dexterities. If the autonomous multi-fingered hand is able to show the dexterities as human hand, it would be a remarkable achievement. Another advantage of the multi-fingered hand is that it can be used for different tasks in harsh, remote place such as space where human access is not only difficult but also sometimes impossible. The importance of the multi-fingered hand is rising in time to meet the demands of the technical world. A few specific areas where multi-fingered hands are being used extensively are as follows:

- **Prosthetic Hand:** This is the most exciting application of a robotic hand which is designed as a replica of the human hand in terms of number of fingers and thumbs. It is also called the ‘Bionic Hand’ and may contain multiple movement possibilities of the fingers and can also be embedded on the human wrist. This bionic hand receives the signal from the brain to allow motion for different tasks. Various bionic hand research projects are

considered as a major research area in the medical field. The Shadow hand is an excellent example of human-like robot hand that is under on-going research in order to establish it as a unique bionic hand [1]. There are other prosthetic hands available which are discussed in the literature [2],[3],[4].

- Industries: The efficiency of machinery can be much higher than a human being in terms of physical labour intensive tasks and gross capacity. The biomimetic hand is well-suited for applications to industrial production where the production is proportional to the labour and hence quite costly. Industries are the most frequent users of biomimetic robot hands to replace low-average human labour intensive tasks at a lower cost. From basic picking and placing tasks to different complex tasks such as welding, shaping and de-burring of multi-shaped objects, robot hands are reliable end-effectors in all types of manufacturing industries. For example, the multifingered Barrett and the DLR hand are used in industries for picking and placing of the objects [5],[6]. Robot hands not only save human energy, but also increases the production in time. Human being cannot always perform at the same level which affects the production time. Moreover, robot can be used 24/7 to speed up the production rate and human are required for observation purpose only.

1.1. Multi-fingered hand features

Classical robotic manipulators are classified broadly into two groups based on the geometry: serial kinematic chain based linkages and parallel kinematic chain based linkages, commonly known as serial and parallel manipulators. The combination of links and pairs without a fixed link is not a mechanism but a kinematic chain. A kinematic chain is also an assemblage of links and joints, interconnected in a way to provide a controlled output motion in response to a supplied input motion. Kinematic chains or mechanisms may be either open or closed. A closed mechanism will have no open attachment points or nodes and may have one or more degrees of freedom. An open mechanism of more than one link will always have more than one degree of freedom, thus requiring as many actuators (motors) as it has degrees of freedom. Robotic systems such as two cooperating arms and multi-fingered hands, which consist of several actively controlled articulated and serial connected linkages, act in parallel in performing a task on an object. Moreover, quite unlike serial manipulators these arms consist of one or more closed kinematic chains in their structure. Unlike parallel manipulators, not only is there more than one actuator in a particular chain, the number of actuators generally exceed the mobility. The number of degrees of freedom of a mechanism is called the mobility of the device. The mobility is the number of input parameters (usually pair variables) that must be independently controlled to bring the device into a particular position. Not only do dualities exist between serial and parallel manipulators, the control problems of these dynamically

constrained systems have been extensively studied. On the other hand, in a robotic hand with multi-fingers which are redundantly actuated closed kinematic chains, while the actuator rates can be uniquely determined from the specified trajectory, the actuator forces and control inputs are under-determined.

A robotic hand involves interaction between the actively controlled fingers and a passive body or object. A first step in their control is to determine the optimum contact conditions between the outer link in a finger or distal phalanx and the body. The contact force distribution is determined from the trajectory errors while assuming the actuators to be ideal. Co-ordination schemes must therefore necessarily avoid any singularities in the configuration when extreme contact force distributions may be generated and also ensure that the contact force distribution is optimum. Singularities, when excessive contact forces are generated, are situations when the mechanism or chain loses a degree of freedom (kinematic singularity) or situations when the contact forces are underdetermined (geometric singularity). In a robotic hand which is already been well designed, such singularities are generally avoided. Moreover by considering a single closed kinematic chain the issue of redundancies can be sidestepped so one can focus on the primary task to determine the optimum contact conditions and the forces associated with contact. A key issue in the related control problem associated with multi-fingered hands is the control of the contact forces. A major difficulty is the fact that the contact forces can only, at best, be controlled indirectly by controlling the phalanxes as no direct control can be affected on them. Salisbury and Craig [7] are one of the first groups of researchers to study these issues.

Multi-fingered robotic hands are used extensively in many robotic applications because of their ability to grasp objects of differing shapes and sizes and to perform manipulation tasks requiring the dexterity of the human hand. Multi-fingered robotic hands are underactuated due to a number of reasons and yet are compact and relatively easy to control. Controllers for these hands may be designed using traditional controller synthesis techniques and validated by simulation. Some of the joints of an underactuated multifingered hand are coupled by couplers such as belt drives (artificial tendons) and gears while the other joints are actuated by servo motors. The number of servo motors used is generally much lower than the number of degrees of freedom. In many underactuated robotic hands, decoupling mechanisms are used to alter the kinematic or dynamic constraint imposed by the coupler just prior to the hand grasping an object. Decoupling facilitates the hand in completely enveloping an object, adapt to its shape and then grasp it with the correct magnitude of the contact forces.

1.2. Grasping and manipulation problem

The problem involving coordination control that can be carried out by the multi-fingered robot hand. The hand is efficient in performing sophisticated cooperative process such as grasping, manipulation, assembling tasks, may end up with difficult issues when organized approach is

not considered to handle it. The issues are: inaccuracies of the mathematical model of the hand and the object, imprecise contact model, errors in execution tasks planning. In general, sophisticated control mechanisms can be implemented on the articulated hand for grasping and manipulation. To match the complexities of the most versatile human hand, many robot hands have been designed over the past decades but have proven not to be completely autonomous in a different scenario. High dimensional complex algorithms are required for these hands as the number of degrees of freedom (DOF) in the hand increases with design complexity. The dexterity of a robot hand can be achieved if the high configuration algorithm problem can be solved. At this point, the best strategy towards these issues may hold the solution with some main aspects, firstly the complete functional architecture of the hand coordinated system and secondly, the software/hardware level infrastructure to support functional architecture and finally the efficiency of the design at the software or hardware level. Over the last two decades, research studies in this area of robotics have been concerned with advanced technical and theoretical development in robotic hand control [8],[9],[10]. This improvement allowed robotics research to re-investigate the coordinated control tasks with regards to grasping and manipulation. The coordinated control refers to the control of the robotic fingers in a coordinated fashion [11]. The idea is to apply advanced control algorithm in a coordinated control framework for grasping and manipulation.

Until the multi-fingered hand was introduced, grasping and manipulation problems remained restricted to gripping and manipulation. The multi-fingered hand illustrates its ability in performing grasping and manipulation tasks at different levels and changed the definition of the object grasping and manipulation activities. Object grasping and manipulation by a robotic hand is regarded as one of the most comprehensive tasks in robotics [12]. The important two functions in this task are to restrain and manipulate the object. The task of restraining the object is called 'fixturing' and manipulating is called "dexterous manipulation". These tasks are employed in industrial robots and can be extremely difficult for moving objects of small scale [13]. Traditional industrial manipulators devised an old fashioned end-effector called the gripper to perform these tasks. The function of the gripper was limited due to a few facts as follows:

1. A non-dexterous design refers to the fact that the gripper can achieve grasp of an object but is unable to manipulate it.
2. A lack of flexibility allows only for the grasp of planar and parallel shaped objects.
3. The dependency on the motion of the body manipulator which it is attached to. This dependency can potentially lead to inaccuracies as it is not practical to move the entire arm for the desired motion in the gripper space.
4. Independently controlled grippers can be chosen to get rid of these problems. However, even if the independent gripper improves motion accuracy, it cannot be used to grasp arbitrary shaped of object due to its non-dexterous design.

The inspiration of investigating multi-fingered robotic hand originates from the need to get over the limitation of existing gripper in performing grasping and manipulation tasks. To illustrate the grasping and manipulation problem with multi-fingered hand, a scenario is presented. Consider a robotic hand which is commanded to grasp an object in a familiar environment. There are some assumptions required prior to that such as known contact locations on the object surface. Contact types are fixed point contact with friction which means that a contact can transmit both normal and tangential force. In the first stage of the problem (also defined as pre-grasp state), the robot hand assumes a pose to grasp the object. To achieve the pre-grasp state, contact points are considered for planning the trajectories from the initial hand position. A position based control law is applied to calculate the joint torques such that the hand reaches the desired contact locations considering dynamic uncertainties and disturbances. These dynamics uncertainties and disturbances are listed as: unmodelled internal robot hand mechanics (tendon/gear mechanics), unknown frictions in the hand joints and the actuators.

The next stage is called the grasp state where the object is grasped based on the contact mechanics. The contact mechanics is developed from the interaction between the robot hand and the object. The contact force calculated from the contact mechanics is the amount of forces required to grasp an object. This force exerts reaction torque on the joints of the hand. Similarly, the force exerted on the object generates motion. This interaction of forces between the hand and the object plays a crucial role in accelerating the object dynamics by the robotic fingers. The precise contact force calculations allow the robotic hand to produce the correct amount of torques so that the object can be manipulated accordingly. If the contact force cannot be controlled to keep it to a minimal value, it may damage the object and also cause harm to the joint actuators due to the excessive amount of forces. For example, a little amount of contact force is enough to grasp an orange. Uncontrolled contact force will result in smashing the orange and producing large amount of torques responsible for motor damage. Therefore, the force control requires the selection of desired contact forces and implements a force controller such that the contact force originated from the position error between the hand and the object, meets the desired forces.

Finally, the problem associated with dynamic control of object manipulation starts with developing a position controller for robot-object combined dynamics such that the object achieves the planned configuration after grasp. This controller should be able to take the contact force into account and control the position of the object and the contact force occurred between robot hand and the object. Generating a robust position control law and deducing a force optimisation algorithm have always been a crucial control requirement when grasping and manipulation is carried out in an unstructured environment. The complex hand and the object structure make it difficult to suggest generalised control requirements for grasping and manipulation. On the other hand, a complicated control algorithm may not be suitable to implement in all types of hands. Therefore, research is still on-going for a global solution to propose highly robust

control algorithms for robotics hands and objects in performing unstructured grasping and manipulation.

Considering grasping and manipulation requirements and multi-fingered hand features, an advanced multi-fingered robotic hand is selected to conduct research in this area. This multi-fingered robotic hand is called the Barrett hand and is designed based on the research first made by Ulrich [14]. The Barrett hand is one of the most effective robotic hands widely used in industries and also for multi-fingered hand research. It is made up of three articulated fingers, each with two phalanxes, which can be actively controlled by a single actuator mounted at the base of each the fingers. The three fingers are connected to the bases by rotary joints [15]. The bases of two of the fingers rotate around the main palm in a relative rotary spreading movement which is actuated by a fourth servo motor. When the distal phalanx makes contact with the object with a specific force, a decoupler uncouples the distal and proximal phalanx by employing a clutch, so that the proximal phalanx can orient itself to envelope the object.

1.3. Recent developments

Grasping and manipulation of the object are two fundamental problems in the study of multi-fingered robotic hand. In the last few decades, major advances have been made in acknowledging attributes of multi-fingered hand towards grasping and manipulation. Depending on different multi-fingered hands, new techniques have been included in the area of grasping and manipulation studies. Literature reviews on multi-fingered grasping and manipulation can be categorised into several sections: kinematics, contact model, grasp planning, grasp control and force optimisation and grasp quality measure.

Robotic fingers move to change an object's initial to a desired position in dexterous manipulation. The manipulation problems are stated with regards to the desired position, the motion response and the characterisation of the forces exerted on the object. All these tasks are related to the hand motion that establishes the control framework for computing the joint torque required to drive the object. The geometric properties required for these are: finger link and object geometries, finger kinematics and contact locations.

To represent the combined hand-object dynamics, some significant kinematic features have been presented in the literature. Kinematics of fingers, contact and the grasp map have been discovered [16],[17],[18],[15],[19],[20]. The grasp map was first determined by Salisbury [16] transforming the hand fingertip forces to the object space ensuring that the object wrench is balanced by the exerted force. Point contact in rigid bodies is first explained by Cai et al [21]. Montana [19, 22, 23] first developed the contact kinematics between robot fingertips and object to derive fundamental contact differential equations. The formulation allowed for the development of the relationship between local fingertips and object frame with their geometric

information known through contact velocities. Kerr [17] presented the kinematics of rolling contact for object manipulation. The study characterised the relationship between joint and object motion whilst considering the rolling contact between the fingertip and the object. A robot-object constraint was introduced to derive the kinematic equations by assuming that the object and fingertip velocities are equal at the contact port. Cole [24, 25] also showed the formulation of rolling contact developed between two arbitrary shaped surfaces.

Contact modelling is also a fundamental tool in multi-fingered grasping and manipulation. The kinematic and dynamic model of a robotic hand can be easily derived using the same techniques applied for a standard manipulator. The same applies for an object which is readily available from rigid body dynamics theories [15]. The combined dynamic modeling of the robot hand-object is not a trivial task as it requires the model of the interaction between them. As stated in section 1.2 that multi-fingered manipulation is an extended version of coordinated operations of multiple arms. The only variance is due to the contact nature between the fingertip and the object surface. Basic cooperating robot arm scenario is presented with a strong contact model for object grasps [26],[27]. In the dexterous manipulation case, the soft fingertip contact model is considered which is more complicated to establish. The effective manipulation refers to the fingertip movement relative to the position held by the object which is related to the contact mechanics between those. Salisbury [16] first presents the interactional modelling problem. In that study, the unit basis twist and wrenches were developed for three typical contacts such as: point contact, point contact with friction and soft finger contact. Point contact only generates a force normal to the surface. The point contact with friction allows force in the tangential direction. The tangential force is proportional to the normal force by Coulomb's friction [15]. A soft finger model also allows moments to resist about the surface normal. Cutkosky [28] first validated the point contact with frictions and soft finger model for grasping case. For soft contact, more details on frictions are presented in further studies [29], [30].

Grasp planning and optimisation refers to the study of how an optimum grasp can be achieved for an object by a multi-fingered hand. Kinematic redundancies of the hand can affect the grasp planned for the same object. Likewise, a grasp plan changes dramatically for different shaped objects. There is a wide selection of grasps possible for an object by the robotic hand. Searching for the best grasp falls into the category of grasp planning study area. This research area also includes a choice of optimal contact location which helps to achieve the force closure of the grasp [31]. Planning of a grasp and its optimisation for different sets of tasks have been presented in the literature [10],[19],[32],[33],[31],[34],[35]. Cole [24] presented an algorithm for grasp planning considering rolling constraints for dexterous manipulation. Li et al. [36] proposed the grasp planning problem in object space considering friction cone and internal forces. This study also defined the transformation between the finger and the object motion. This transformation characterises the multi-fingered hand workspace and also provides the evaluation criteria for hand design. The study by Hong et. al. [37] presented multifinger

grasps for two and three-dimensional objects with small friction. The complete planning and coordinated control of multiple manipulators with rolling contacts is discussed by Paljung and Kumar [38]. This research also included optimal contact point determination on both the robot and the object frame for specified task configuration. The control law is based on the nonlinear feedback that linearises and decouples the system which also simultaneously controls the system state trajectory. Finger-object coupled kinematics in configuration spaces are determined by Montana [23]. The study considered the combined system as large closed kinematic chain combining small chain for finger and contact point. Research on rolling motion of arbitrary objects appeared in Bichhi's work [39]. This research discussed the planning and the controlled ability of manipulation through rolling contact. The issues of optimal object grasping by a robotic hand are expressed in Li and Sastri's work [31]. This study also evaluated grasp quality measure properties. Nyugen [32, 33] proposed algorithms for planar optimal grasp and also discussed how to achieve force closure stable grasp. Montana [22] used differential geometry to derive the time evolution history of the contact point position on the object surface when no external force is exerted on it and also measured the contact stability of the two-fingered hand grasp. Bicchi et al [9] introduced multi-robot dynamics for object manipulation. They studied grasp in velocity space considering different kinematics.

Control algorithms for multi-fingered coordinated problem with fixed or point contact have been substantially evaluated in [40],[27],[41],[24],[42],[43],[44],[45],[46]. The flexibility, force control consistency, control parameter optimisation, singularities and control structure of multi-fingered hand are introduced by Salisbury and Craig [7] where the control framework is developed and allowed for direct position and force control of an object after grasp for a desired motion configuration. Based on the task model, Li et al [36, 47] established dual concepts of grasp stability and manipulability and also expressed the structured framework to measure grasp quality. That study formulated a generalised computed torque algorithm in an object's workspace for grasping and manipulation of the multi-fingered robot hand with point contact. Cole et al. [24] focused on the control of arbitrary shaped object manipulation by multi-fingered hand for rolling contact. Su and Stepanenco [48] discuss adaptive sliding mode based multi-fingered coordinated control for unstructured multi-fingered manipulation in their studies. Adaptive mechanisms were used to estimate the mass of the object and robotic hand. From this, the adaptive control framework was proposed for a manipulation problem considering unknown dynamics. Adaptive control structures for co-ordinated control are presented in [49],[50]. The global aspects of dexterous motion planning for 3D objects by multi-fingered robotic hand are addressed by Cherif and Gupta [51].

Optimisation of grasping forces and grasp stability are another fundamental area in the study of multi-fingered grasping and manipulation. The stability of a grasp can be determined by the force closure properties [32]. A grasp is called a force closure grasp when forces of the hand fingers balance the external forces exerted on the grasped object [15]. The grasp can be

synthesized by locating fingertips associated with the contact model to establish force closure grasp. Another research issue is to analyse a grasp that is categorised by contact points and a contact model to assess the force-closure properties [15],[41].

A number of acceptable conditions of force closure are proved by Salisbury and Roth [16]. They showed that, at the contact points, contact wrenches are produced through contact forces that span the total wrench space positively. This means that the wrench frame origin always lies in the convex hull of the contact wrenches. Some other force closure conditions for three fingered hands are classified by Ponce et al. [52]. They computed the grasps in two dimensions only by following those conditions though the grasp. Nyugen [32] developed an algorithm to assess the force closure properties of two fingered grasps. Bicchi [53] analysed the force closure properties by adopting Lyapunov's direct method. He also provided some conditions to test the properties along with linear programming computation.

Grasp force optimisation considering the force closure grasp is a major concern in multi-fingered manipulation. Many algorithms are proposed for computing optimal grasping forces for the multi-fingered grasping [26],[54],[55],[56],[57],[58]. Kumar and Waldron [58] proposed force distribution with sub optimal algorithms. Yoshiakawa et al [8] formulated a structure to compute internal grasping force to solve three-fingered grasp problem. Nakamura [59] considered the optimisation problem as a nonlinear programming problem and solved it applying the Lagrange multiplier. The first step in solving these algorithms is to linearise the friction cone constraints and apply linear programming. Buss et al [55] first indicated that specific symmetric matrices with positive definite properties are equivalent to the nonlinear coulomb friction constraints. From this observation, the force optimization problem was formulated with symmetric positive definite matrices with specified cost functions under Reimannian manifold [60]. The solutions are found based on the gradient flow projection method. The features of this algorithm are that it solves the non-linearised friction cone constraints with optimal control method. Han et al [61] applied friction cone constraint as Linear Matrix Inequality (LMI) problem and developed the grasp as convex optimisation set.

Control procedures for multi-fingered manipulations have been surveyed in [13],[62]. The foundations required for control implementations are thoroughly detailed in [15] and are classified into some particular control methods. These control methods are: model based control, hybrid position or force control, stiffness control and impedance control.

Model based control is achieved by following the system dynamics. For multi-fingered grasping, the system refers to the hand and the object considering contact constraints. The advantages of this type of control is that the system's non-linear characteristics can be fully compensated by selecting computed torque or inverse dynamics based control laws to achieve the asymptotic tracking of an object's centre of mass. When the system dynamic model has uncertainties, this control technique is feasible to run the simulation, but not useful in real applications.

To track the object centre of mass while maintaining a fixed type contact between the object and the finger, an inverse dynamics based control law is developed by Cole [24]. The contact slippage is ignored by ensuring that the contact force always lies within the friction cone constraints during the execution of the desired tasks. This control law also allows the control of internal forces in order to satisfy the friction cone constraints which are specified by the null space of the grasp matrix. This framework is also considered for the rolling contact. With force feedback, a computed torque control is implemented [63]. The joint torques are computed in two subspaces: before and after the grasp state. The first set of torques drive the hand joints to reach the object contact locations by the fingertips before grasp. The second sets of torques are computed based on the force exerted on the fingertips by the object after grasp. In this framework, the desired internal forces are computed by optimising the friction cone so that the object's desired configuration can be achieved without slipping. Chen and Zribi [64], proposed a computed torque law considering point contact for both rolling and sliding motion.

The control of both the position and the forces are required to execute the manipulation tasks such that the slippage of the fingers can be prevented. When decoupling is considered, the position and the force can be controlled concurrently. Nagai et al [65] considered the finger forces as the addition of two orthogonal components: grasping force (to maintain friction constraint) and the manipulation force (to induce object motion). A feedback control law is proposed by this study to decouple and linearise the system with subject to the sets of grasping and manipulation variables. The Proportional-Integral-Derivative (PID) controller is applied to control the object position and orientation and the Proportional-Integral (PI) controller is considered to satisfy the friction cone constraints for avoiding slippage. Another feedback linearisation method [66] is developed where system dynamic equation are presented in a workspace in order to avoid the complicated inverse kinematics problem and to provide a differential equation set through algebraic constraints. The input-output linearisation method based on the state space is discussed by Isildori [67]. This investigation is considered to decouple the force and position space whilst the pole placement method is employed to achieve the desired system behaviour. In a study by Dougeri et al [68], another hybrid feedback controller is implemented as a serial combination of a control law. Each law is designated to regulate the object position and orientation and to realize grasp stability. Remond et al [69] proposes a decentralized hybrid scheme. Each finger is controlled independently by this approach and in addition, a task planner computes the desired force and position in finger space to track contact force and object motion. This research work also proposed adaptive solution for uncertain system parameters. Hilhorst et al [70], developed the control framework in two stages: first, the mass of the object is determined by the nonlinear adaptive method and secondly, the estimated parameters are used to optimise the object's velocity and determine the optimal internal forces satisfying the friction constraints. Taking the sliding contact and unknown system dynamics into account, a coordinated control framework is proposed for multi-fingered manipulation [71]. The convergence of object motion and contact forces are achieved asymptotically by adaptation and proved in the

sense of Lyapunov [60].

A grasp is defined stable when object motion and contact force error occurring from external disturbances reduce over time. Contact model specifies the contact forces and object motion through frame transformation. The contact stiffness determines how hard or soft the grasp will be. The contact stiffness a grasp refers to the geometric map which transforms the differential displacement of the hand fingertips to the change of force exerted on the fingertips. Grasp stiffness properties based on contact geometry, contact conditions and finger mechanical properties have been presented in the literature [72]. In this work, the equation that combines the object's motion and grasping forces is linearised to compute the grasp stiffness. The control law proposed here imposed a desired stiffness to achieve the grasp by adjusting the control gain of the hand fingers. Stiffness matrix errors are defined and actuator gains are updated consequently. A decentralized stiffness method for multi-fingered grasping is also presented in the literature [73]. Cartesian based stiffness control is evaluated by Son et al [74] where the controller can regulate the stiffness of the grasped object. In a Cartesian stiffness control structure, the controller is implemented to measure the compliance error. Reaction forces are computed to compensate the compliance error. This reaction force is exerted on the object and can be transformed to the finger and joint forces by the grasp matrix and finger Jacobian respectively. The finger and joint level stiffness control problem has been solved in [75]. First, the desired compliance in an operational space is decomposed into a hand fingertip frame. Then, the compliance characteristic expressed in the fingertip frame is decomposed in the hand joint frame to allow the independent joint control of the hand. This study also mentioned the limitation of the hand mechanics and the drawback of force and tactile sensors used in the hand. A complete planning and control framework for dexterous manipulation is presented in the literature [76]. The control module inputs are the commanded object motion and the associated contact force in this case. Desired compliance is considered at hand finger space, transformed into finger velocities. These velocities are controlled by the joint level controllers.

The interaction between the robot hand and an object can be directly controlled by the Impedance control method. In this method, when an object is grasped by the hand, the couple hand object system is considered as second order mass-spring damper system. The impedance control goal is to achieve the desired behavior of the hand object system. Schneider et al [77] proposed an impedance based control system for grasping an object by cooperative manipulators. The desired impedance is achieved at the object level with this control mode. In Stramigioli's [78] study, passivity based impedance control is implemented which is allowed to be used in free or grasped state. The concept of a virtual object is addressed in this work. The virtual object is considered to be linked to the hand finger by a spring of variable length and linked to a virtual point on the other side by another spring. The contact point compliance can be regulated by adjusting the spring parameters. Biagotti et al [79] discussed an impedance based scheme based on the inner-outer feedback loop. The inner loop maintains the position by rejecting

the disturbances while the outer loop acts as an admittance controller to impose mass spring damper characteristics to the hand fingers. Another impedance technique has been mentioned in [80] which combines a grasp force optimisation algorithm also to execute different states of the object manipulation as well as re-grasping.

In the last decades, significant advances were made in noticing attributes of multifingered manipulation. The complete surveys on grasping and manipulation features became available in the literature. Shimoga [81] published a wide survey on functionality of grasp. Bichhi and Kumar [12] reviewed the multi-fingered robotic grasping and contact evolution. Bichhi [13] established the state of the art of the robotic hand and summarised its progress and pointed out future aspects of multi-fingered hand in the society. Okamura et al [82] made research analysis in dexterous multi-fingered manipulation. Walker [83] presented analysis, design and control of the articulated robot hand.

1.4. Motivations, aim and objectives

This research is inspired by various applications of the robotic hand. These include: the usage of the ‘Bionic Hand’ as a human hand replacement, the applications in cost-effective industrial production and applications in remote and harsh landscapes where human access is very limited. The prospective applications of the robotic hand have always been an interesting area to me. These interests encouraged me to undertake the control challenges of the nonlinear robotic hand for accomplishing different applications. It would be a great achievement to control the robotic hands for the applications previously mentioned in order to overcome the limited access of the human being. The control challenges of the robotic hand are related to grasping and manipulation. This is another interesting area with multiple control design challenges. Different types of robotics hands, object, contact constraints and model can entirely change the control objectives of grasping and manipulation. The varying characteristics of the hand, the object and the model present formidable challenges for controller design, analysis and implementation for grasping and manipulation.

The aim of the current research is to develop an intelligent control framework for grasping and manipulation of an object by a multi-fingered robotic hand in order to adapt to those varying characteristics and accomplish the manipulation. The objectives of the research works are illustrated in a few steps, as follows:

- Development of a mathematical model of a robotic hand, an object and the contact model between the hand and the object.
- Specification of the control problem of the grasping and manipulation tasks by the given hand. The control specification will describe the types of control input and the output solutions of the robot-object system.

- Development of an intelligent control framework considering the varying nature of the robotic hand and the objects, contact constraints and system uncertainties. The intelligent controller refers to the control features which will allow the articulated hand to grasp and manipulate the object with more accuracy and less hand and object information.
- Analysis and performance comparison of the developed controllers with the existing one.

1.5. Contributions

In this research, the multi-fingered Barrett hand is considered to solve grasping and manipulation tasks. The hand is modelled based on the Lagrangian method [15]. Joint and Cartesian space control solutions are first obtained through classical algorithms and secondly higher order sliding mode control algorithm is applied. To achieve robustness against uncertainties, higher order sliding based adaptive impedance controller is developed for grasping and manipulation of an object by the multi-fingered Barrett hand. The main contributions are as follows:

- A Lagrangian based mathematical model of the multi-fingered Barrett hand is developed. To develop the model, firstly the forward and inverse kinematics of the hand are determined. The workspace of the Barrett hand is computed from the inverse kinematics. The kinematics provided by the manufacturers has mathematical faults that are corrected in the model. The SimMechanics based Barrett hand is developed using the CAD design of the real hand [84]. The SimMechanics model allows the complete animation based framework for grasping and manipulation.
- An object mathematical model is developed using the Newton-Euler modelling method. The interaction between the Barrett fingertip and object is modelled considering point contacts with friction by following Coulomb's model [15]. This provides the ability to analyse the compliance behaviour of the complete control system. Then, the interactional model is transformed to the hand and the object space by calculating the Jacobian. The contact model is established between the Barrett hand and the object and the contact force is measured. The physical platform of the control environment is also developed through SimMechanics.
- A Computed torque control (CTC) based controller is implemented for the position control of the Barrett hand. Both the joint and the workspace based controllers are implemented to see the motion performances of the joint and the fingertip. The simulations are generated for SimMechanics based Barrett Hand and the mathematical model. The hand model has been validated from the tracking results. System errors originating from physical and mathematical model are compared and the controller performance is analysed based on the errors.

- Classical controllers are unable to deal with the structured and unstructured uncertainties of the system. Three adaptive higher order sliding mode controllers are proposed for their robust nature against uncertainties. The controller implemented for position control of the Barrett finger under noise and noise free cases. These controllers improved the chattering performance occurs in the classical sliding mode controller. The controller performances are found better compared to the computed torque method.
- A position and force based impedance controllers are implemented for a planar and a Barrett finger. The advantages and disadvantages of both control methods are discussed from the simulated results. An adaptive based impedance method is proposed and implemented to track the contact forces with unknown object properties. These properties are estimated using an adaptive mechanism to generate online based input trajectories to achieve the desired contact force. The adaptation based impedance controller enhanced the performance of the contact force by an online estimation mechanism. The controller stabilities are proved using Lyapunov's direct method.
- A computed torque and two stiffness based controllers are implemented for grasping and manipulation of an object by the Barrett hand. Computed torque controller is developed to accomplish grasp and manipulate an object in the object space. Two natural and a Stiffness controllers are developed as the extension of natural control law to achieve less tracking errors. Stiffness based controllers showed better performance compared to the others.
- To solve the grasping and manipulation under uncertain dynamics and unknown object properties, an adaptive higher order sliding mode controller is proposed and implemented to track the desired motion of the object and the desired contact force to grasp it. In this case, an arbitrary desired force is chosen to meet the actual grasping force. Results found from the simulations are compared with stiffness based control laws and the feasibility of the proposed controllers are discussed.

1.6. Outline of the thesis

The chapter which has just been presented provides an introduction to the thesis. It covers the background to the main problem addressed in thesis, the recent developments in relation to the control of grasping and manipulation by robotic hands, the challenges and motivations behind the thesis, the significant and main contributions of the thesis, a brief outline of the entire thesis and a list of specific publications arising from the research.

Chapter 2 concerns the dynamics and modelling of the Barrett hand as well the characterisation of the object as a free body, its kinematic description and the dynamics of unrestrained motion. The chapter includes a discussion of the forward and inverse kinematics of the Barrett hand,

the finger dynamics where each finger is modelled as a two-link manipulator and the dynamics of the three-finger and palm assembly. The chapter is concluded with a section on how the hand may be modelled within the environment of SimMechanics, a modelling tool provided by Matlab & Simulink.

Chapter 3 defines the basic control problem related to the hand. This includes the position control of the palm-finger assembly which constitutes the hand, until the fingers make contact with the object, followed by the control of the grasp. Basic ideas behind Lyapunov stability and control, input-output stability and computed torque control are presented.

Chapter 4 introduces the concept of sliding mode control and the application of sliding mode control to the control of the posture of the fingers. Also considered are the adaptive implementations of sliding mode control when the dynamic properties of the fingers are not known precisely as well as the application of higher order sliding mode control theories.

Chapter 5 concerns the basic ideas of grasping and manipulation, such as the development of contact forces, the analysis of the mechanics of contact and the indirect control of the forces of contact.

The focus of chapter 6 is the theories of grasping and manipulation, and the application and implementation of classical techniques to the control of grasping. The need for planning the grasp, the properties of the grasp, the concept of the grasp map and the importance of force-closure are introduced in the chapter. Also presented in the chapter is the combine dynamics of the hand and the unconstrained object it is grasping.

Chapter 7 concerns the application of the sliding mode and adaptive sliding mode control to the grasp control problem. In particular the application of the sliding mode controller synthesis techniques with respect to the control and manipulation of an object following a grasp are presented in detail.

The thesis concludes with a discussion of the primary results followed by the enunciation of the author's thoughts on the future directions of the contributions of the research, in chapter 8.

1.7. Publications

Publications made from this research are outlined below.

1. Md Rakibul Hasan, R Vepa, M H Shaheed and H Huijberts, "Modelling and Control of the Barrett Hand for Grasping," *Computer Modelling and Simulation (UKSim)*, UKSim 15th International Conference on, 2013.
2. Md Rakibul Hasan, A Rahideh and M H Shaheed, "Modeling and interactional control of the multifingered hand," *Automation and Computing (ICAC)*, 19th International Conference on, 2013.

3. Md Rakibul Hasan, R Vepa, H Huijberts and M H Shaheed, “Modeling and Adaptive Higher-Order Sliding-Mode Control of the Multi-fingered Barrett Hand,” Submitted to the *Journal of Franklin Institute*.
4. Md Rakibul Hasan, R Vepa and M H Shaheed, “Impedance-based Higher Order Sliding Mode Control for Grasping and Manipulation,” accepted and due to be published in the *proceedings of The Sixth International Conference on Advances in System Simulation, SIMUL 2014*.

2. Modelling of the Barrett hand and the hand-object system

2.1. Introduction

Multi-fingered robotic hands introduce enormous potential in robot industries as efficient tools to be used by different manipulators for various tasks [12]. Robot arms or manipulators perform tasks such as picking and placing, cutting, grasping and manipulating with the help of an end-effector. An ineffective end-effector limits the productivity of the whole robotic arm. Most grippers have fewer degrees of freedom, are customised for individual tasks and have only a fixed direction of motion. In general, gripper customisation is required for every variation in shape, orientation and pose of the end-effector frame with respect to the task space frame. These limitations reduce the productivity in industrial environment. The integration of the multi-fingered robotic hand as an end-effector is a great alternative to conventional grippers. The robotic hand provides a new approach to tasks such as assembling machinery, handling materials, sampling collection in remote places and spaces.

A multi-fingered hand is dexterous compared to handling and assembling done by grippers and always has fewer chances of slipping which causes damage to an object. Most of the multi-fingered hands available in industries are configurable and programmable for all types of workspace activities. The programmable capabilities of these hands provide more scopes to improve the control abilities for more complex grasping and manipulation tasks. In this chapter, a multi-fingered hand "The Barrett hand" is considered for modelling as prerequisite of implementing the advanced controllers. These controllers are easily programmable to assess the performance of the hand in different manipulation environment.

2.2. The Barrett hand description

The BH8-series *Barrett hand*TM is a multi-fingered grasper with great flexibility to grasp objects of different shapes and sizes [5]. The hand can be used with most of the manipulators available in the industry. It is a completely self-contained system with low weight. The hand

2.2 The Barrett hand description

consists of three fingers F_1, F_2, F_3 and a palm (Figure 2.1). Each finger has two revolute joints and consists of three links $l_1 = 0.07m, l_2 = 0.05m, l_3 = 0.05m$. The masses of the links are: $m_1 = 0.162kg, m_2 = 0.0579kg, m_3 = 0.0342kg$. The first revolute joints of all fingers are called finger based joints and the second joints are called fingertip joints. The fingers F_1 and F_2 can rotate around the palm in the x direction which gives similar dexterity to the human thumbs, but F_3 is secured to the palm base. The spread movement of the fingers F_1 and F_2 give extra degrees of freedom (DOF) for the hand operation. The hand is controlled by four DC brush-less Servo Motors. The two joints of each finger are driven by a single motor placed on the base while the spread movement of the fingers F_1 and F_2 is achieved by a fourth motor. The initial angular positions and motion ranges of the joints are given in Table 2.1.

Table 2.1.: Static joint positions of the Barrett hand and their motion range.

Joint type	Initial angular position q_{oi}	Motion Range q_{im}
Fingerbase joint	2.46°	140°
Fingertip joint	40°	45°
Spread joint	0.0001°	180°

The initial three dimensional (3D) posture of the Barrett hand computer aided design (CAD) model is presented in Figure 2.1. In this diagram, the dimensions of the finger links l_i , initial angles q_{oi} and joint motion variables q_i are given as notation. The base frame C_b is attached as a reference frame of the hand finger kinematics. The palm frame is defined as C_p and the base coordinate of all fingers are expressed as C_{fb_i} . The spread angle is shown as α_i .

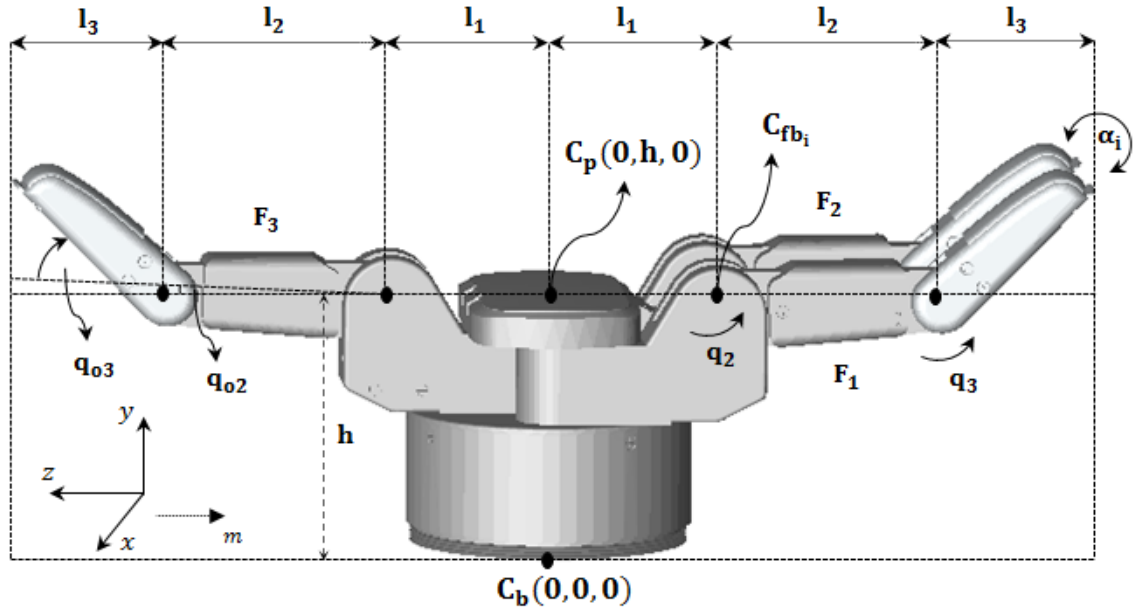


Figure 2.1.: The Barrett hand CAD model in its initial position.

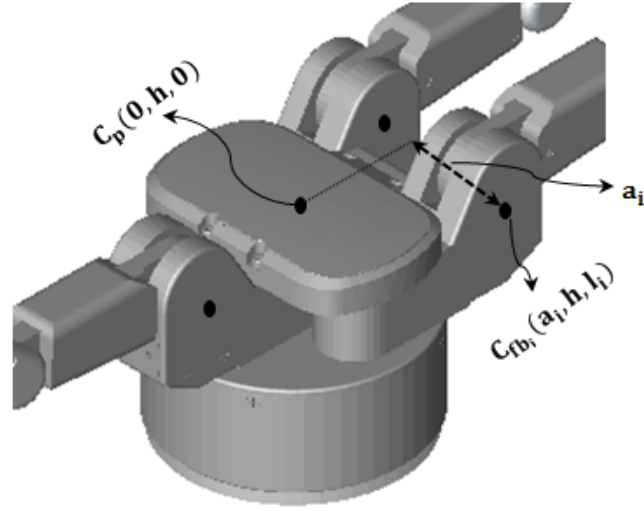


Figure 2.2.: Side view of The Barrett hand CAD model.

2.3. Hand Kinematics

Robot manipulator kinematics describe the relationship between the joints of the links and the resulting motion of the robot bodies. The robotic hand is a composed form of the fingers placed on the base. Each finger is imagined as a single rigid open chain manipulator without any constraint. Each finger may have one or multiple joints for producing motions. Joint mechanisms have several forms such as revolute, prismatic and helical. Forward and inverse kinematics are calculated depending on the types of joint mechanisms available in the hand finger.

2.3.1. Forward Kinematics

Forward kinematics expresses the relationship between the Cartesian space of the robot body relative to the joint position. Given a joint angle for each link, this derives the Cartesian position of the link end. The Barrett hand forward kinematics are determined according to Figure 2.1. This relation for the hand can be mathematically stated as,

$$X = f(Q), \quad (2.1)$$

where $X \in \mathbb{R}^3$ is the workspace position and $Q \in [q_{oi}, q_{imax}]$ are given as joint configuration. All joints are revolute and angles are measured using the right-handed coordinate system. Figure 2.3 gives the forward kinematics structure of the Barrett hand.

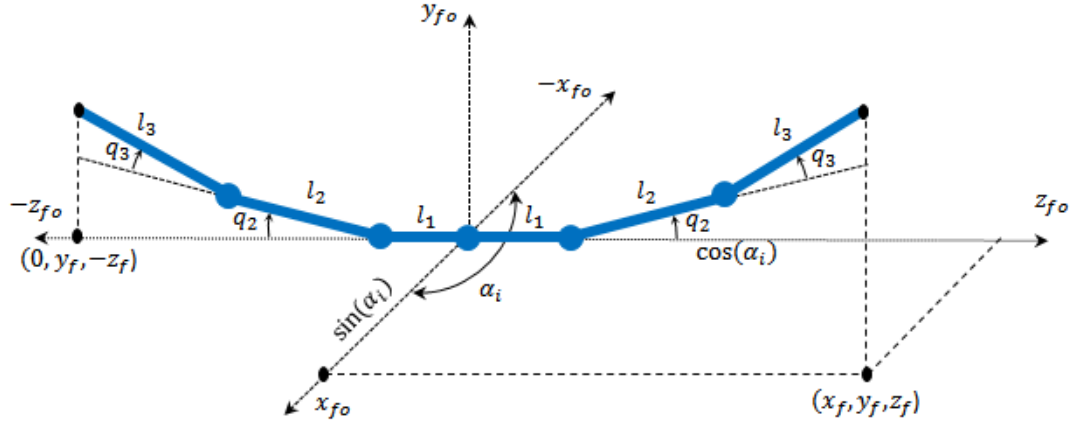


Figure 2.3.: The Barrett hand forward kinematics.

For fingers F_1 and F_2 , the joint rotations are counter clockwise and the rotation is clockwise for finger F_3 . Let $(x, y, z)_{F,i}$ be the Cartesian fingertip position of the fingers and q_i is the joint variable of the fingers. The term $q_i (i = 1, 2)$ can be chosen as desired angle to calculate the fingertip coordinates for any finger independently. The forward kinematics of the fingers F_1 and F_2 are expressed in (2.2) and (2.3) as,

$$\begin{aligned} x_{f1} &= (l_2 \cos(q_2 + q_{o2}) + l_3 \cos(q_2 + q_3 + q_{o2} + q_{o3})) \sin(\alpha_1) \\ y_{f1} &= h + l_2 \sin(q_2 + q_{o2}) + l_3 \sin(q_2 + q_3 + q_{o2} + q_{o3}) \\ z_{f1} &= l_1 + (l_2 \cos(q_2 + q_{o2}) + l_2 \cos(q_2 + q_3 + q_{o2} + q_{o3})) \cos(\alpha_1) \end{aligned} \quad (2.2)$$

$$\begin{aligned} x_{f2} &= (l_2 \cos(q_2 + q_{o2}) + l_3 \cos(q_2 + q_3 + q_{o2} + q_{o3})) \sin(\alpha_2) \\ y_{f2} &= h + l_2 \sin(q_2 + q_{o2}) + l_3 \sin(q_2 + q_3 + q_{o2} + q_{o3}) \\ z_{f2} &= l_1 + (l_2 \cos(q_2 + q_{o2}) + l_2 \cos(q_2 + q_3 + q_{o2} + q_{o3})) \cos(\alpha_2) \end{aligned} \quad (2.3)$$

and the forward kinematics of the finger F_3 are below,

$$\begin{aligned} y_f &= h + l_2 \sin(q_2 + q_{o2}) + l_2 \sin(q_2 + q_3 + q_{o2} + q_{o3}) \\ z_f &= l_1 + l_2 \cos(q_2 + q_{o2}) + l_2 \cos(q_2 + q_2 + q_{o2} + q_{o3}) \end{aligned} \quad (2.4)$$

In equation (2.4), due to the absence of the spread angle α_i , the joint rotation exists only on $z - y$ axis. All fingers of the hand is used to reach any desired location in the workspace using the forward kinematics.

2.3.2. Inverse Kinematics

Inverse kinematic problems refers to achieving the joint angle configurations for a given Cartesian location in the workspace. For a given fingertip position and orientation, the problem is to

find the joint variables of each robot link. Solving for the joint variables is not simple because it may have multiple solutions. Firstly, consider a finger of the Barrett Hand in the planar case to illustrate the inverse kinematic problem. The inverse problem is to solve the equations (2.2) and (2.3) for q_1 , q_2 and q_3 in order to achieve the desired fingertip position (x_f, y_f, z_f) . The problem can be solved using the polar coordinate definition of (r, ϕ) [85]. For a desired position (x_f, y_f, z_f) and the initial base position (x_o, y_o, z_o) of the Barrett finger F_i , the joint angles of the fingers are calculated using the trigonometric relationship. The spread angles for the fingers F_1 and F_2 are,

$$q_{sp,i} = j \left(\tan^{-1} \left(\frac{(x_f - x_o)}{z_f} \right) \right), \quad (2.5)$$

where $(i = 1, 2)$ and $j = \{-1, 1, 0\}$ for the fingers $[F_1, F_2, F_3]$. The joint angles of finger F_3 are obtained from trigonometry as,

$$\begin{aligned} q_{2,F_3} &= \tan^{-1} \left(\frac{q}{p} \right) - q_{o2} \\ q_{1,F_3} &= \tan^{-1} \left(\frac{q}{p} \right) - \tan^{-1} \left(\frac{n_1}{n_2} \right) - q_{o1} \end{aligned} \quad (2.6)$$

The joint angles of finger F_i ($i = 1, 2$) is calculated as,

$$\begin{aligned} q_{2,F_i} &= \tan^{-1} \left(\frac{q}{p} \right) - q_{o2} \\ q_{1,F_i} &= \pi - \tan^{-1} \left(\frac{q}{p} \right) - \tan^{-1} \left(\frac{n_1}{n_2} \right) - q_{o1} \end{aligned} \quad (2.7)$$

where the terms $\cos q_3 = (p^2 + q^2 - l_2 - l_3)/(2l_1l_2)$; $p = (z_f - z_o)$; $q = (y_f - y_o)$; $n_1 = l_2 + l_3 \sin(q_3)$; $n_2 = l_3(\sin q_3)$. When the configurations of the joints are outside the range of the Barrett hand, the solutions are not found. Therefore, it is important to know the location of the hand in the workspace depending on the joint range q_{max} . The workspace is required for planning and executing a manipulation task by the hand. The set of workspace ranges of the Barrett hand can be defined as,

$$R_{(X,Y,Z)} = \{r(q) : q \in Q\} \subset \mathbb{R}^3, \quad (2.8)$$

where $r(q) : Q \rightarrow \mathbb{R}^3$ is position component achieved from the forward kinematics and Q is the complete set of joint configurations.

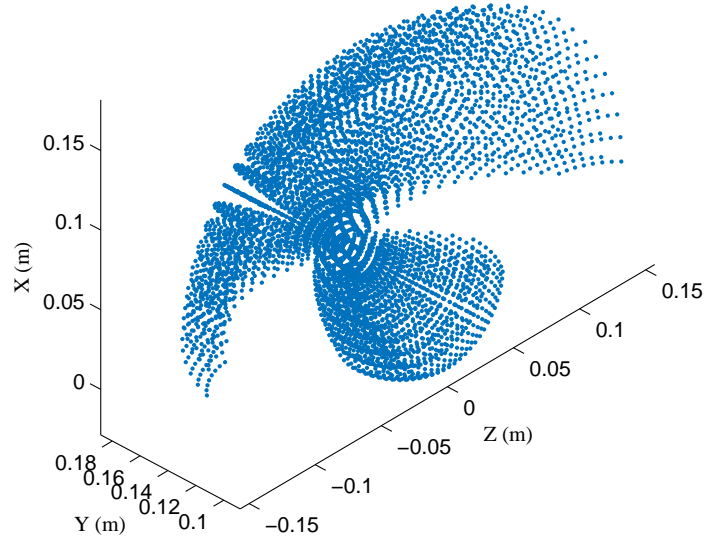


Figure 2.4.: The workspace range of finger F_1 for joint angle q_i and α_1 calculation.

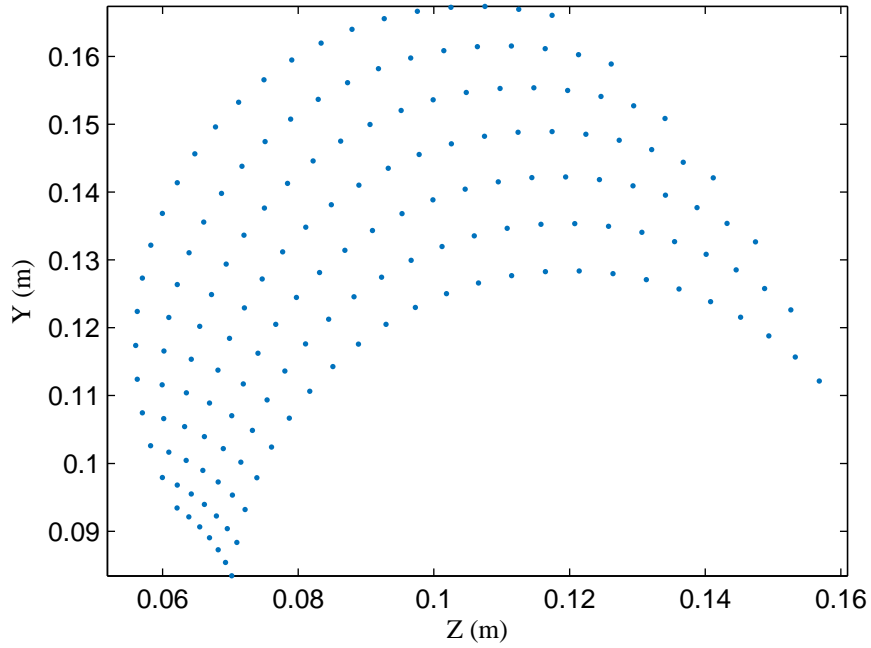


Figure 2.5.: The workspace range of finger F_3 for joint angle q_i calculation.

In Figure 2.4, the workspace range of F_1 is shown in three dimensions and the range of F_1 and F_2 are mirror symmetric. For finger F_3 , the workspace range is limited to two dimensions only due to the absence of spread angle α_i (See Figure 2.6). The workspace coordinate range for each finger is summarised in Table 2.2.

Table 2.2.: Workspace range of the Barrett hand.

Finger Name	$R_X (m)$	$R_Y (m)$	$R_Z (m)$
F_1	$[-0.02, 0.18]$	$[0.1, 0.18]$	$[-0.15, 0.15]$
F_2	$[0.02, -0.18]$	$[0.1, 0.18]$	$[-0.15, 0.15]$
F_3	0	$[0.1, 0.18]$	$[0.05, 0.15]$

2.4. Barrett hand dynamics

The forward and inverse kinematic models of the hand described in the section 2.3 shows the relation between the joint and the workspace of the robot. In this section, the robotic hand dynamics will be discussed. The robotic hand dynamics are a set of finger dynamics which exhibit the motion responses of the hand followed by the commanded input. The dynamics for each finger is seen as an open chain manipulator and is driven by the actuator available at the joint links of the finger. Actuators are generally driven through the motor torques. The actuator torques can be considered directly as control input to the robot dynamics. The torques are computed by the controller to achieve the desired position of the joints.

2.4.1. Finger dynamics

The finger dynamics is a set of second order ordinary differential equations containing nonlinearities. These equations are formed based on the kinematic and inertial properties of each finger. There are many methods available to determine the equations of motions of a mechanical system. For dynamic analysis of a robot system, the Lagrangian based method is suited for computation and is therefore used in this chapter to derive the finger dynamics [15]. Using this technique, the potential and kinetic energies of the system are calculated from the forward kinematics. The Lagrangian approach is applied to derive the equations of motions for all three fingers of the Barrett hand. The only difference is the existence of spread angle α_i for fingers F_1 and F_2 . Hence, the dynamics of finger F_1 and F_2 are identical and the finger dynamics of F_3 is restricted to two dimension only. The kinetic energy of each links of finger F_1 and F_3 is derived to form the hand dynamics. Consider the finger F_1 first to express the kinematics for each links. The corresponding velocities of F_1 are found for second joint link l_2 from (2.2) by differentiating with respect to time as follows,

$$\begin{aligned} \dot{x}_1 &= l_2 \dot{\alpha} c_{\alpha 1} c_2 - l_2 \dot{q}_2 s_{\alpha 1} s_2 \\ \dot{y}_1 &= l_2 \dot{q}_2 c_2 \\ \dot{z}_1 &= -l_2 \dot{\alpha} s_{\alpha 1} c_2 - l_2 \dot{q}_2 c_{\alpha 1} s_2 \end{aligned} \quad , \quad (2.9)$$

and the F_1 fingertip velocities are shown below,

$$\begin{aligned}\dot{x}_2 &= \dot{\alpha}_1(l_2 c_{\alpha 1} c_2 + l_3 c_{\alpha 1} c_{23}) - l_2 \dot{q}_2 s_{\alpha 1} s_2 - l_3 \dot{q}_3 s_{\alpha 1} s_{23} \\ \dot{y}_2 &= l_2 \dot{q}_2 c_2 + l_3 \dot{q}_3 c_{23} \\ \dot{z}_2 &= -\dot{\alpha}_1(l_2 c_2 s_{\alpha 1} + l_3 c_{23} s_{\alpha 1}) - l_2 \dot{q}_2 c_{\alpha 1} s_2 - l_3 \dot{q}_3 c_{\alpha 1} s_{23}\end{aligned}, \quad (2.10)$$

where $s_{i/\alpha} = \sin(q_{i/\alpha})$, $s_{ij} = \sin(q_i + q_j)$, $s_{i-j} = \sin(q_i - q_j)$ and $c_{i/\alpha} = \cos(q_{i/\alpha})$, $c_{ij} = \cos(q_i + q_j)$, $c_{i-j} = \cos(q_i - q_j)$.

2.4.2. Energy calculation for *Lagrangian*

From (2.10), the kinetic energy T of finger F_1 can be calculated from the following expressions,

$$T_{F_1}(q, \dot{q}, \alpha, \dot{\alpha}_1) = 0.5m_2(\dot{x}_1^2 + \dot{y}_1^2 + \dot{z}_1^2) + 0.5m_3(\dot{x}_2^2 + \dot{y}_2^2 + \dot{z}_2^2), \quad (2.11)$$

and the potential energy V is calculated by the terms below,

$$V_{F_1}(q, \alpha) = m_2 g y_1 + m_3 g y_2. \quad (2.12)$$

The difference between the kinetic (T) and the potential (V) energy of the system is called the *Lagrangian* L and is shown below,

$$\begin{aligned}L_{F_1}(q, \dot{q}, \alpha, \dot{\alpha}_1) &= T(q, \dot{q}, \alpha, \dot{\alpha}_1) - V(q, \alpha) \\ &= 0.5m_2(\dot{x}_1^2 + \dot{y}_1^2 + \dot{z}_1^2) + 0.5m_3(\dot{x}_2^2 + \dot{y}_2^2 + \dot{z}_2^2) - m_2 g y_1 - m_3 g y_2.\end{aligned} \quad (2.13)$$

Lagrange's equations for deriving the equation of motions of the hand finger with generalised coordinates $(q, \alpha) \in \mathbb{R}^n$ and the *Lagrangian* L are given as,

$$\frac{d}{dt} \left(\frac{\delta L}{\delta(\dot{q}_i, \dot{\alpha}_i)} \right) - \frac{\delta L}{\delta(q_i, \alpha_i)} = \tau_i \quad i = 1, 2, \dots, n, \quad (2.14)$$

where τ_i is the external forces acting on the system coordinate.

2.4.3. Dynamics of the fingers F_1 and F_2

Inserting (2.9) into (2.14) yields three equations of motions for finger F_1 which can be expressed in matrix form as follows,

$$M_{ij}\ddot{q} + C_i(q, \dot{q}) + G_i(q) = \tau_f, \quad (2.15)$$

where $M(q, \alpha)$ is the $n \times n$ symmetric, positive definite inertia matrix of the finger, $C(q, \dot{q})$ is the $n \times 1$ vector of centrifugal and Coriolis terms, $G(q)$ is the $n \times 1$ gravity torque vector

($i, j = 1, 2, \dots, n$; n is the number of degrees of freedom). The inertia matrix and gravity torque are related to each finger's joint positions. The centrifugal and Coriolis terms are related to the joint positions and the velocities. Equation (2.15) can be explicitly showed as,

$$\begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} \ddot{q}_2 \\ \ddot{q}_3 \\ \ddot{\alpha}_1 \end{bmatrix} + \begin{bmatrix} C_1(\dot{q}_2, \dot{q}_3, \dot{\alpha}_1) \\ C_2(\dot{q}_2, \dot{q}_3, \dot{\alpha}_1) \\ C_3(\dot{q}_2, \dot{q}_3, \dot{\alpha}_1) \end{bmatrix} + \begin{bmatrix} G_1(q) \\ G_2(q) \\ G_3(q) \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}. \quad (2.16)$$

The elements of the inertia matrix M_{F_1} are given below,

$$M_{11} = m_2 l_2^2 + m_3 l_3^2 + m_3 l_2 l_3 \cos(q_2 - q_3),$$

$$M_{12} = m_3 l_2 l_3 \cos(q_2 - q_3),$$

$$M_{21} = m_3 l_2 l_3 \cos(q_2 - q_3) + m_3 l_3^2,$$

$$M_{22} = m_3 l_3^2,$$

$$M_{13} = M_{23} = M_{31} = M_{32} = 0,$$

$$M_{33} = m_2 l_2^2 c_2^2 + m_3 l_2^2 c_2^2 + m_2 l_3^2 c_3^2 + 2l_2 l_3 m_3 c_2 c_3,$$

and the vectors of Coriolis and centrifugal terms C_{F_1} are as follows,

$$C_1(\dot{q}_2, \dot{q}_3, \dot{\alpha}_1) = m_3 l_2 l_3 s_{2-3} (\dot{q}_2 + \dot{q}_3 + \dot{\alpha}_1^2) + (m_2 + m_3) l_2^2 \sin(2q_2) + m_3 l_2 l_3 s_{23},$$

$$C_2(\dot{q}_2, \dot{q}_3, \dot{\alpha}_1) = 0.5 \dot{\alpha}_1^2 (m_3 l_2 l_3 s_{23} - m_3 l_2 l_3 s_{2-3} - m_3 l_2 l_3 s_{2-3} + m_3 l_3^2 \sin(2q_3)) \\ - m_3 l_2 l_3 \dot{q}_2^2 s_{2-3},$$

$$C_3(\dot{q}_2, \dot{q}_3, \dot{\alpha}_1) = -l_2 q_3 \dot{\alpha}_1 (m_3 l_2 \sin(2q_2) + m_3 l_2 \sin(2q_2) + 2m_3 l_3 c_3 s_2) \\ - (m_3 l_3^2 \sin(2q_3) + 2m_3 l_2 l_3 c_2 s_3) (\dot{q}_2 + \dot{q}_3) \dot{\alpha}_1,$$

and the gravitational terms G_{F_1} are given as,

$$G(q_1) = g (m_2 l_2 c_{23} + m_3 l_2 c_{23}),$$

$$G(q_2) = g (m_2 l_3 c_2),$$

$$G(q_3) = 0.$$

2.4.4. Dynamics of the finger F_3

Similarly, the velocities of the corresponding second link l_2 of the finger F_3 on the $z-y$ plane are calculated below,

$$\begin{aligned} \dot{y}_1 &= l_2 \dot{q}_2 c_2 \\ \dot{z}_1 &= -l_2 \dot{q}_2 s_2 \end{aligned}, \quad (2.17)$$

and the F_3 fingertip link l_3 velocities are given as,

$$\begin{aligned} \dot{y}_2 &= 2l_2 \dot{q}_2 c_2 + l_3 \dot{q}_3 c_{23} \\ \dot{z}_2 &= -2l_2 \dot{q}_2 s_2 - l_3 \dot{q}_3 s_{23} \end{aligned}. \quad (2.18)$$

The kinetic (T) and potential (V) energy for the finger F_3 is calculated similarly as follows,

$$T_{F_3}(q, \dot{q}, \alpha, \dot{\alpha}_1) = 0.5m_2 (\dot{z}_1^2 + \dot{y}_1^2) + 0.5m_3 (\dot{z}_2^2 + \dot{y}_2^2) \quad (2.19)$$

$$V_{F_3}(q, \alpha) = g (m_2 y_1 + m_3 y_2). \quad (2.20)$$

From (2.19) and (2.20), the *Lagrangian* is defined as,

$$L_{F_3} = 0.5m_2 (\dot{z}_1^2 + \dot{y}_1^2) + 0.5m_3 (\dot{z}_2^2 + \dot{y}_2^2) - m_2 g y_1 - m_3 g y_2. \quad (2.21)$$

Inserting (2.19) into *Lagrangian* equation (2.14) provides motion equation for finger F_3 . The dynamics form is similar to (2.16) and is expressed in matrix form as shown below,

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} C_1(\dot{q}_2, \dot{q}_3) \\ C_2(\dot{q}_2, \dot{q}_3) \end{bmatrix} + \begin{bmatrix} G_1(q) \\ G_2(q) \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}, \quad (2.22)$$

where

$$M_{11} = (m_2 + m_3)l_2 l_2 + m_3 l_3 l_3 + 2m_3 l_2 l_3 c_{23} + m_3 l_3 l_3 + m_3 l_2 l_3 c_{23}$$

$$M_{12} = m_3 l_3 l_3 + m_3 l_2 l_3 c_{23}$$

$$M_{21} = m_3 l_3^2 + m_3 l_2 l_3 c_{23} + m_3 l_3^2$$

$$M_{22} = m_3 l_3^2$$

$$C_1(\dot{q}_2, \dot{q}_3) = -m_3 l_2 l_3 s_{23} (\dot{q}_2^2 - (\dot{q}_2 + \dot{q}_3)^2)$$

$$C_2(\dot{q}_2, \dot{q}_3) = m_3 l_2 l_3 s_{23} \dot{q}_2^2$$

$$G_1(q) = g ((m_2 + m_3)l_2 c_2 + m_3 l_3 c_{23})$$

$$G_2(q) = g (m_3 l_3 c_{23})$$

2.4.5. Dynamics of the complete hand

From (2.16) and (2.22), the complete Barrett Hand dynamics can be represented as,

$$M_{f_i} \ddot{Q}_{f_i} + C_{f_i}(\dot{q}_i, \dot{\alpha}_i) + G_{f_i} = \tau_{f_i}, \quad (2.23)$$

and the matrix form can be shown as,

$$\begin{bmatrix} M_{f_1} & \cdots & \cdots \\ \cdots & M_{f_2} & \cdots \\ \cdots & \cdots & M_{f_3} \end{bmatrix} \begin{bmatrix} \ddot{Q}_{f_1} \\ \ddot{Q}_{f_2} \\ \ddot{Q}_{f_3} \end{bmatrix} + \begin{bmatrix} C_{f_1}(q, \dot{q}, \alpha, \dot{\alpha}) \\ C_{f_2}(q, \dot{q}, \alpha, \dot{\alpha}) \\ C_{f_3}(q, \dot{q}) \end{bmatrix} + \begin{bmatrix} G_{f_1} \\ G_{f_2} \\ G_{f_3} \end{bmatrix} = \begin{bmatrix} \tau_{f_1} \\ \tau_{f_2} \\ \tau_{f_3} \end{bmatrix}, \quad (2.24)$$

where the hand inertia matrix $M_f \in \mathbb{R}^{8 \times 9}$, hand Coriolis and centrifugal vector terms $C_f \in \mathbb{R}^{8 \times 1}$ and hand gravitational vectors $G_f \in \mathbb{R}^{8 \times 1}$ are used to represent the hand dynamics. The input to the hand is the joint torque vector $\tau_f \in \mathbb{R}^{8 \times 1}$ and the term $Q_f \in (q, \alpha)$ is the vector set of joint configurations. To solve for \ddot{Q}_{F_i} , (2.23) can be represented as,

$$\begin{bmatrix} \ddot{Q}_{f_1} \\ \ddot{Q}_{f_2} \\ \ddot{Q}_{f_3} \end{bmatrix} = \begin{bmatrix} M_{f_1} & \cdots & \cdots \\ \cdots & M_{f_2} & \cdots \\ \cdots & \cdots & M_{f_3} \end{bmatrix}^{-1} \begin{bmatrix} \tau_{f_1} \\ \tau_{f_2} \\ \tau_{f_3} \end{bmatrix} - \left(\begin{bmatrix} C_{f_1}(q, \dot{q}, \alpha, \dot{\alpha}) \\ C_{f_2}(q, \dot{q}, \alpha, \dot{\alpha}) \\ C_{f_3}(q, \dot{q}) \end{bmatrix} + \begin{bmatrix} G_{f_1} \\ G_{f_2} \\ G_{f_3} \end{bmatrix} \right). \quad (2.25)$$

Equation (2.25) contains a total of eight equations of motion which is solved to reach the desired joint configurations of the hand fingers; this is referred to as joint space dynamics.

2.4.6. Structural properties of hand dynamics

Several dynamic properties can be constructed from the above dynamics as presented in (2.23) which are essential for developing controller. The properties are given below,

1. The matrix $M_f(q, \alpha)$ is symmetric and positive definite. It is uniformly bounded and satisfies,

$$w_1 I_n \leq M_f(q, \alpha) \leq w_2 I_n \quad \forall q, \alpha \in \mathbb{R}^{n \times n}, \quad (2.26)$$

where w_1 and w_2 are positive constant and $I_n \in \mathbb{R}^{n \times n}$ is the identity matrix. The matrix M_f exists and is positive definite too.

2. For a given finger dynamics, the matrix $C_f(q, \dot{q}, \alpha, \dot{\alpha})$ is not always unique but $C_f(q, \dot{q}, \alpha, \dot{\alpha})\dot{Q}_f$ is always unique. The term $C_f(q, \alpha, 0) = 0$ for all vectors $(q, \alpha) \in \mathbb{R}^n$. The term $(\dot{M}_f - 2C_f) \in \mathbb{R}^{n \times n}$ is related to the inertia matrix M_f and satisfies,

$$x^T [\dot{M}_f - 2C_f] x = 0 \quad \forall q, \dot{q}, \alpha, \dot{\alpha}, x \in \mathbb{R}^n, \quad (2.27)$$

and as a matter of fact that, $\dot{M}_f - 2C_f$ is skew symmetric matrix.

3. The dynamic equation of (2.23) can be linearly parametrized as,

$$Y_f(q_d, \dot{q}_d, \ddot{q}_d) = M_f(q_d)\ddot{q}_d + C_f(\dot{q}_d, q_d)\dot{q}_d + G_f(q_d), \quad (2.28)$$

where $q \in \mathbb{R}^p$ contains system parameters that are unknown. The desired regression matrix is expressed as $Y_f(q_d, \dot{q}_d, \ddot{q}_d) \in \mathbb{R}^{n \times p}$ which contains known bounded functions of the desired link position $q_d(t)$, velocity $\dot{q}_d(t)$, and the acceleration \ddot{q}_d trajectory. It is assumed that all trajectories are bounded functions of time.

2.4.7. Ratio dynamics and torque switch

It is noted from the Barrett fingers that only one servo motor is used to control two revolute joints of each finger F_i [12]. The motor drives the first joint and the second joint is driven by the first one internal pulley mechanism. The inner joint q_2 first moves by the given torque τ . The outer joint q_3 is coupled to the previous joint and moves one third of the q_2 . Then, the torque τ is transferred to the second joint by the *torqueswitch* mechanism. This is a conditional mechanism activated by the reaction torque exerted by the object when the contacts are made by the Barrett fingers. The mechanism is explained in a flow chart given in Figure 2.7.

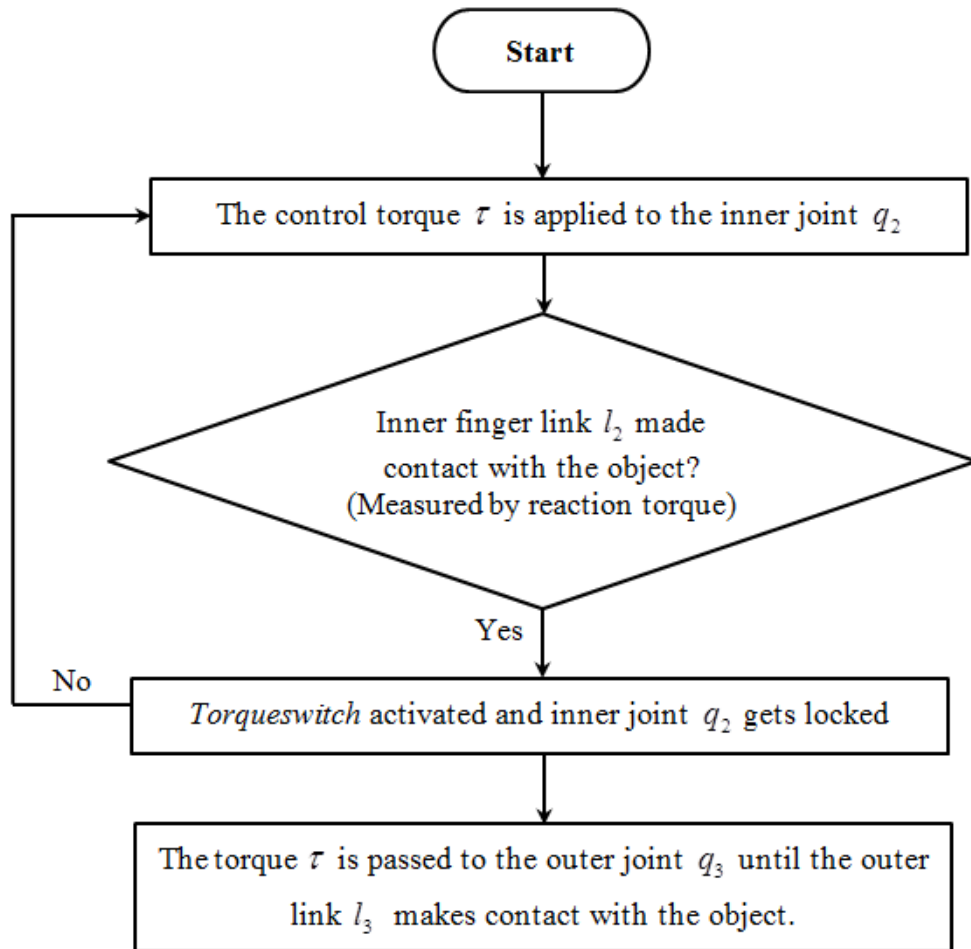


Figure 2.6.: The motion based conditional mechanism of *torqueswitch*.

To derive the dynamics behind the coupling mechanism with one third ratios, recall the dynam-

ics of finger F_3 from (2.22) in linear form as,

$$\begin{aligned} M_{11}\ddot{q}_2 + M_{12}\ddot{q}_3 + C_1(q, \dot{q}) + G_1(q) &= \tau_1 \\ M_{21}\ddot{q}_2 + M_{22}\ddot{q}_3 + C_2(\dot{q}, \dot{q}) + G_2(q) &= \tau_2 \end{aligned} \quad (2.29)$$

when the inner joint of the finger is generated by the motor torque, it is governed by the first equation of (2.29). At that time, the outer joint behaves q_2 as follows,

$$\begin{aligned} q_2 &= \sigma q_1 \\ \dot{q}_2 &= \sigma \dot{q}_1, \\ \ddot{q}_2 &= \sigma \ddot{q}_1 \end{aligned} \quad (2.30)$$

where, m is the one third ratio angular scale followed by the outer joint. Substituting (2.30) into (2.29) yields,

$$(M_{21} + \sigma M_{22})\ddot{q}_2 + C_2(\sigma q, \sigma \dot{q}) + G_2(q) = \tau_2. \quad (2.31)$$

The term $(M_{21} + \sigma M_{22})$ in the above equation reflects the scaled movement of outer joint q_3 by following the initial conditions given in (2.30). Therefore, τ_2 is designed based on the dynamics of (2.31). When the inner joint links of fingers make contact with the object, the *torqueswitch* is activated (See Figure 2.7). The activation is related to the reaction torque τ_d . At that instant, the only joint torque τ is transferred to the outer joint q_3 . In that case, the inner joint becomes locked by the internal mechanism which indicates $\ddot{q}_1 = 0$ and $u_1 = 0$ and the outer joint is governed by equation (2.31).

2.5. Workspace dynamics

When a desired configuration of the robotic hand is given in the workspace as a function of time, inverse kinematics is used to calculate the joint angles. The inverse kinematics calculation process is time consuming and require powerful computers for computing the desired angle q_d at a suitable rate. The feedback gain selections in the joint space are not trivial as the desired task is presented in Cartesian coordinates. It is therefore desirable to formulate the dynamics of the hand directly in the workspace coordinates. This eliminates the problem of slow inverse kinematics calculation and the control gains can be directly selected in the workspace coordinates. The general finger dynamics from (2.23) is as follows,

$$M_f\ddot{q}_f + C_f(q, \alpha, \dot{q}, \dot{\alpha}) + G_f = \tau_f. \quad (2.32)$$

Jacobian mapping J is used to express the above dynamics in Cartesian coordinates. The Jacobian is related to the joint and the Cartesian coordinates by the following,

$$\dot{X}_f = J_f(q)\dot{q} \quad J_f(q) = \frac{\delta f}{\delta q}, \quad (2.33)$$

where the term $X_f \in \mathbb{R}^{3 \times 1}$ represents the Cartesian coordinates of the fingertip. From (2.33), \dot{q} and \ddot{q} are found as,

$$\dot{q} = J_f^{-1}\dot{X}_f + f \quad \ddot{q} = J_f^{-1}\ddot{X} + \frac{d}{dt}(J_f^{-1})\dot{X}. \quad (2.34)$$

Substitute \dot{q} and \ddot{q} from (2.34) into equation (2.32) and by pre-multiplying J^{-T} yields,

$$J_f^{-T}M_fJ_f^{-1}\ddot{X}_f + \left(J_f^{-T}C(q,\dot{q})J_f^{-1} + J_f^{-T}M_f(q)\frac{d}{dt}(J_f^{-1}) \right) \dot{X}_f + J_f^{-T}G(q) = J_f^{-T}\tau, \quad (2.35)$$

where

$$\begin{aligned} M_x &= J^{-T}M_fJ^{-1} \\ C_x &= J_f^{-T} \left(C(q,\dot{q})J_f^{-1} + M_f(q)\frac{d}{dt}(J_f^{-1}) \right) \\ G_x &= J_f^{-T}G_f \\ F &= J^{-T}\tau \end{aligned} \quad (2.36)$$

From (2.35) and (2.36), the workspace dynamics can be expressed as,

$$M_x(q)\ddot{X}_f + C_x(q,\dot{q})\dot{X}_f + G_x(q) = F_x. \quad (2.37)$$

It is seen from the workspace dynamics that, both M_x and C_x are dependent on the joint variables q and \dot{q} . The matrices M_x, C_x and G_x are called the effective parameters of the workspace dynamics. During simulation or in a real environment, the fingertip vector X_f is calculated by the forward kinematics because joint motion q can be measured directly. The structural properties of the workspace dynamics are similar to the one described in section 2.4.6. Therefore, (2.37) satisfies the properties like below,

1. $M_x(q)$ is symmetric and positive definite.
2. $\dot{M}_x - 2C_x \in \mathbb{R}^{n \times n}$ is skew-symmetric matrix.

Since Jacobian J is invertible, $M_x(q)$ is symmetric and positive definite. A calculation of the term $(\dot{M}_x - 2C_x)$ can be used to show that this is skew-symmetric [86].

2.6. Physical model of the Barrett hand

For the optimization of any system in terms of control or design, physical system and the software need to be developed together. Physical modelling provides the opportunity to combine system level physical domain and control system in a single environment. The physical model

can be reused with physical ports and contains readily available mechanical components. The physical system responses can be extended to 3D visualisation during simulation.

2.6.1. SimMechanics for physical modelling

SimMechanics is a physical modelling software of the MATLAB & Simulink platform which can be used for mechanical simulation. This software enables users to develop and simulate rigid multibody mechanical systems by following the basic principles of mechanics: usage of standard Newtonian dynamics of forces and torques. Unlike the simulink blocks which represent the mathematical calculations, SimMechanics provide mechanical blocks (joints, actuators, sensors, springs, dampers etc) to model a system. SimMechanics can be used seamlessly with Simulink for controller development. After the model is developed, there are different modes of simulation available for analyzing the system. Amongst these, forward and inverse modes are used for modelling and simulation of the system. The SimMechanics based input and output configurations are as follows:

- Forward Dynamics: This calculates the motion of the system resulting from the applied forces or torques and constraints.
- Inverse Dynamics: Required forces or torques are determined to produce the motion for open loop system in this mode.

2.6.2. Conversion of Barrett CAD into SimMechanics

To develop the complete Barrett hand model in SimMechanics, the solidworks based CAD model is used [84]. SimMechanics have the facilities to import the CAD model designed in CAD softwares such as Solidworks and Invent. The conversion processes are presented in the literature [87]. To help understanding the SimMechanics implementation, the SimMechanics model of One link robot with revolute joint is shown in Figure 2.7.

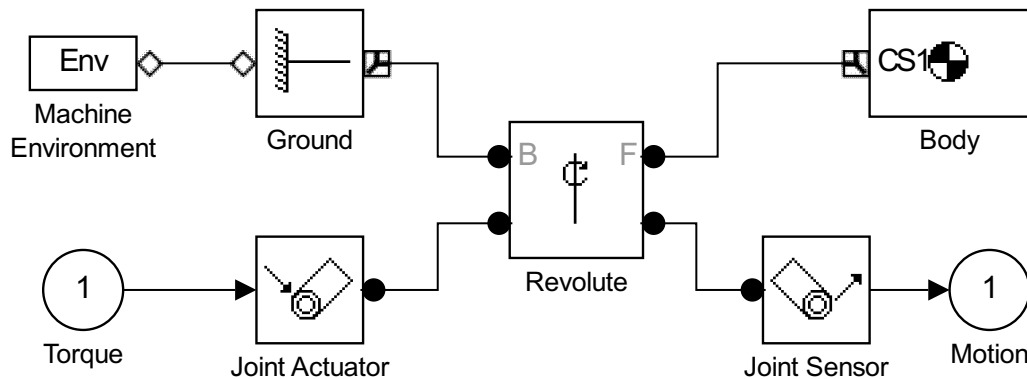


Figure 2.7.: Single rigid robot link with one DOF revolute joint.

2.6 Physical model of the Barrett hand

In Figure 2.7, the block "Machine environment" defines the analysis mode for simulation and tolerance parameters. The 'ground' block defines the fixed reference frame. 'Revolute' block represents the revolute joint type. This is attached to the 'ground'. An actuator is connected to the block which receives input signals (torque) and a sensor is connected to receive the output signals (motion). The 'body' blocks defines the rigid body by mass m , inertia I and the coordinate of the origin (for position and orientation). The model does not require to derive the mathematical equations behind the mechanism. The input torque calculated from the controller can directly be applied to the rigid link. The rigid link animation is shown in Figure 2.8. Depending on the torque, the link moves with respect to the centre (indicated by the black dot) where the joint is attached. More complex robotics models with constraints can be implemented using SimMechanics with visualisation features.

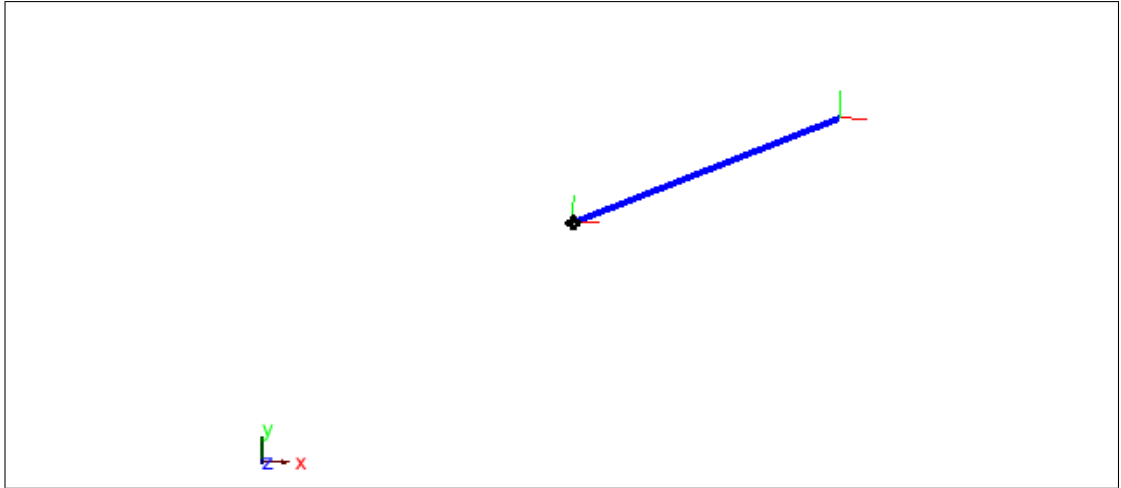


Figure 2.8.: Single rigid robot link with one DOF revolute joint.

The basic principles of SimMechanics procedures discussed above are used to develop the SimMechanics based Barrett hand model. The SimMechanics model of the Barrett hand is presented in Figure 2.9. Seven input (green) channels are available to receive the input signals (torques). Twelve output channels are connected to the joint sensors to measure the joint position (q) and the velocity \dot{q} . Three output channels are connected to the body sensors which compute the fingertip locations in the Cartesian coordinate. The hand model obtained through simulation is presented in Figure 2.10. Each of the bodies is defined subject to the position and the orientation coordinates as shown in the figure. The model for the other two fingers F_1 and F_2 have a similar model format. All SimMechanics model parts are attached to construct the complete shape of the Barrett hand given in Figure 2.12.

2.6 Physical model of the Barrett hand

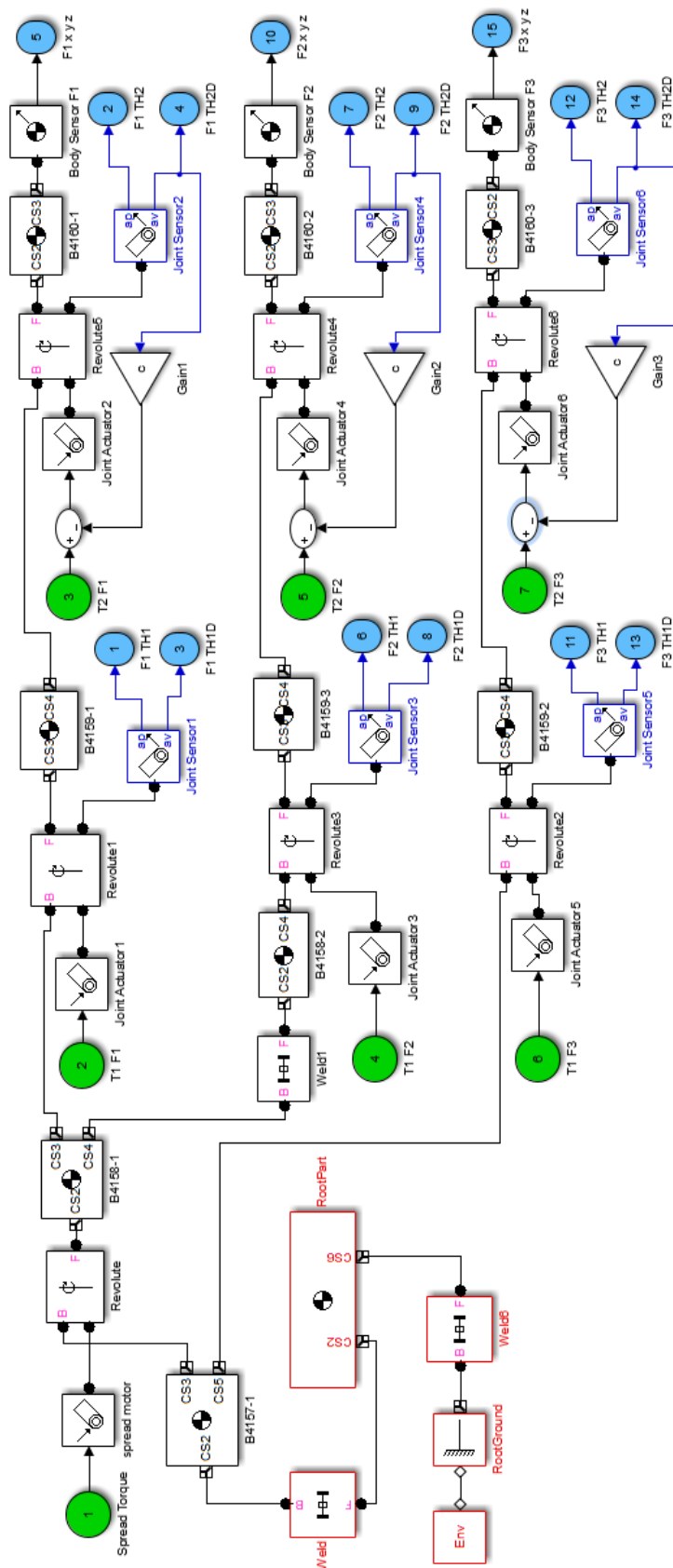


Figure 2.9.: SimMechanics model architecture of the Barrett hand.

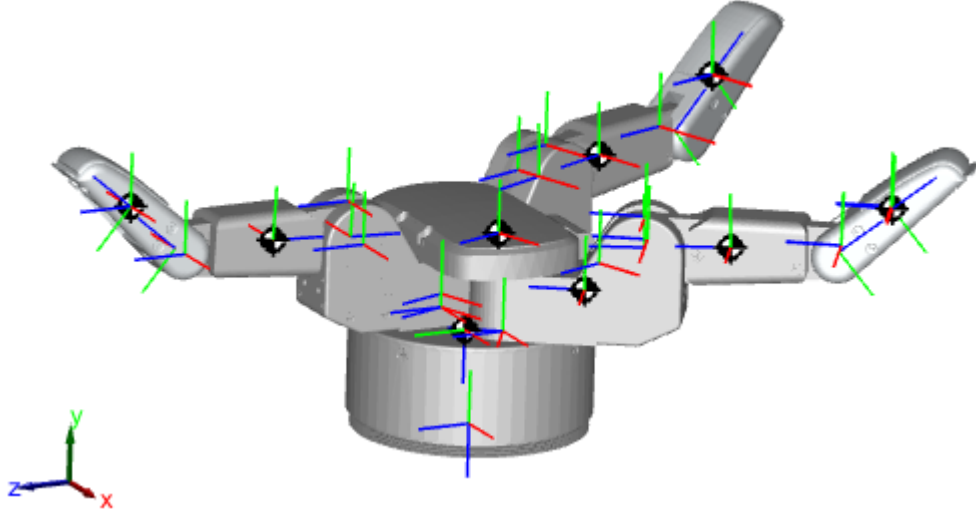


Figure 2.10.: Physical SimMechanics model of the three fingered Barrett hand.

2.7. Object Dynamics

The *Lagrangian* method is used to derive the dynamics of an object. A rigid body configuration in $P = SE(3)$ (where, $P \in \mathbb{R}^6$ is the motion vector) is not directly applicable to be represented by the Lagrangian method but a local parametrisation of the configuration space is used to solve the problem. Euler angles are employed to parametrise the orientation of the rigid body. The global presentation of the dynamics of a rigid body can be presented in terms of the external force and the torques considering the Euler configuration. Let $P = (p, \omega) \in \mathbb{R}^6$ be the coordinate frame attached to the centre of mass of the rigid object where p is the translational and ω is the orientation motion. The term f is the force applied at the centre of the mass of the object. All parameters are relative to an inertial frame. The mass of the rigid body is constant therefore the translational object motion at the centre of the mass can be expressed with Newton's law as,

$$f = m\ddot{p}. \quad (2.38)$$

The angular motion of the object can also be derived separately. Consider the rotational motion ω of the rigid body about a point subject to the external torque τ . The angular momentum with respect to the inertial frame is given by,

$$I_b = RIR^T, \quad (2.39)$$

where R is the transformation frame relative to the inertial frame. Then, the angular motion equation becomes,

$$\tau = \frac{d}{dt}(I_b \omega). \quad (2.40)$$

From (2.40), the final Euler's equation is expressed as,

$$I_b \dot{\omega} + \omega \times I_b \omega = \tau. \quad (2.41)$$

Equation (2.38) and (2.41) represent the dynamics of the rigid body subject to force and torque applied at the center of the body. The force and torque related vector is normally represented with respect to inertial frame, not relative to object frame at the center of the mass. To express the object dynamics with respect to the object frame, twists and wrenches are applied [15]. Therefore, the equations of motion for the object in terms of external wrench F_o employed at the center of the mass is shown in the matrix form below,

$$\begin{bmatrix} m_o I & 0 \\ 0 & I_b \end{bmatrix} \begin{bmatrix} \dot{v}^b \\ \dot{\omega}^b \end{bmatrix} + \begin{bmatrix} \omega^b \times m v^b \\ \omega^b \times I_b \omega^b \end{bmatrix} = F^b, \quad (2.42)$$

where $m_o I \in \mathbb{R}^{3 \times 3}$ is object mass matrix, v^b is the object velocity and ω^b is the body angular velocity. The orientation based inertia tensor I_b is defined by the Euler angle. Euler sequence $Z - Y - Z$ is used in this case. With the Euler angle, the rotation from one frame to another can be visualized as a sequence of three rotations about the base vectors. Each rotation is made through an Euler angle about a specified axis. In the case of the $Z - Y - Z$ Euler angle sequence, the first rotation is about the third axis, through the angle, followed by the second rotation about the second axis through the angle and then followed by a final rotation again about the third axis by the angle. Applying the Euler sequence, the relation between the angular velocity vector of the object $\omega_o = [p \ q \ r]^T$ and the Euler vector $[\phi \ \theta \ \psi]^T$ can be represented as,

$$\omega_o = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\theta} s \psi - \dot{\phi} s \theta \cos \psi \\ \dot{\theta} c \psi + \dot{\phi} s \theta s \psi \\ \dot{\psi} + \dot{\phi} \cos \theta \end{bmatrix} = \begin{bmatrix} -s \theta \cos \psi & s \psi & 0 \\ s \theta s \psi & c \psi & 0 \\ c \theta & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \quad (2.43)$$

From (2.43), the inverse relation is found as,

$$\begin{aligned} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} &= \begin{bmatrix} -s \theta \cos \psi & s \psi & 0 \\ s \theta s \psi & c \psi & 0 \\ c \theta & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \\ &= -\frac{1}{s \theta} \begin{bmatrix} \cos \psi & -s \psi & 0 \\ -s \theta s \psi & -s \theta c \psi & 0 \\ -c \theta c \psi & s \psi c \theta & -s \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \end{aligned} \quad (2.44)$$

The orientation based dynamics come from the bottom row of (2.42) which can be represented as,

$$\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \frac{d}{dt} \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} (I_{zz} - I_{yy})qr \\ (I_{xx} - I_{zz})pr \\ (I_{yy} - I_{xx})pq \end{bmatrix} = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix}. \quad (2.45)$$

The term $d/dt([p \ q \ r]^T)$ is found as,

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} p \\ q \\ r \end{bmatrix} &= \begin{bmatrix} -s\theta \cos\psi & s\psi & 0 \\ s\theta s\psi & c\psi & 0 \\ c\theta & 0 & 1 \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \\ &+ \left(\begin{bmatrix} s\theta s\psi & c\psi & 0 \\ s\theta c\psi & -s\psi & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{\psi} + \begin{bmatrix} -c\theta \cos\psi & 0 & 0 \\ c\theta s\psi & 0 & 0 \\ -s\theta & 0 & 0 \end{bmatrix} \dot{\theta} \right) \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \end{aligned} \quad (2.46)$$

and the terms qr , pr and pq can be expressed as,

$$\begin{aligned} \begin{bmatrix} qr \\ pr \\ pq \end{bmatrix} &= \begin{bmatrix} 0 & \psi + \phi c\theta & 0 \\ 0 & 0 & \dot{\theta} s\psi - \dot{\phi} s\theta \cos\psi \\ \dot{\theta} c\psi + \dot{\phi} s\theta s\psi & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} s\psi - \dot{\phi} s\theta \cos\psi \\ \dot{\theta} c\psi + \dot{\phi} s\theta s\psi \\ \dot{\psi} + \dot{\phi} \cos\theta \end{bmatrix} \\ &= \begin{bmatrix} (\psi + \phi \cos\theta)(\dot{\theta} \cos\psi + \dot{\phi} \sin\theta \sin\psi) \\ (\psi + \phi \cos\theta)(\dot{\theta} \sin\psi - \dot{\phi} \sin\theta \cos\psi) \\ (\dot{\theta} \sin\psi - \dot{\phi} \sin\theta \cos\psi)(\dot{\theta} \cos\psi + \dot{\phi} \sin\theta \sin\psi) \end{bmatrix}. \end{aligned} \quad (2.47)$$

Now, substituting (2.46) and (2.47) into (2.45) yields,

$$\begin{aligned} \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} -s\theta c\psi & s\psi & 0 \\ s\theta s\psi & c\psi & 0 \\ c\theta & 0 & 1 \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} &= \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} \\ &- \left(\begin{bmatrix} s\theta s\psi & c\psi & 0 \\ s\theta c\psi & -s\psi & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{\psi} + \begin{bmatrix} -c\theta c\psi & 0 & 0 \\ c\theta s\psi & 0 & 0 \\ -s\theta & 0 & 0 \end{bmatrix} \dot{\theta} \right) \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \\ &- \begin{bmatrix} (I_{zz} - I_{yy})(\psi + \phi \cos\theta)(\dot{\theta} \cos\psi + \dot{\phi} \sin\theta \sin\psi) \\ (I_{xx} - I_{zz})(\psi + \phi \cos\theta)(\dot{\theta} \sin\psi - \dot{\phi} \sin\theta \cos\psi) \\ (I_{yy} - I_{xx})(\dot{\theta} \sin\psi - \dot{\phi} \sin\theta \cos\psi)(\dot{\theta} \cos\psi + \dot{\phi} \sin\theta \sin\psi) \end{bmatrix}. \end{aligned} \quad (2.48)$$

Equation (2.48) is orientation based dynamics of the object and refers to the bottom row of the (2.42). Let be the local parametrization of (2.42) gives the following,

$$M_o \ddot{X} + C_o(X, \dot{X}) \dot{X} = F_o \quad (2.49)$$

where, $X \in \mathbb{R}^6$ is the object position and orientation vector and the full object mass matrix $M_o \in \mathbb{R}^{6 \times 6}$ can be expressed as,

$$\begin{pmatrix} m_{o1} & 0 & 0 & 0 & 0 & 0 \\ 0 & m_{o2} & 0 & 0 & 0 & 0 \\ 0 & 0 & m_{o3} & 0 & 0 & 0 \\ 0 & 0 & 0 & -I_{xx}s\theta c\psi & s\psi & 0 \\ 0 & 0 & 0 & I_{yy}s\theta s\psi & I_{yy}c\psi & 0 \\ 0 & 0 & 0 & I_{zz}c\theta & 0 & I_{zz} \end{pmatrix} \quad (2.50)$$

and the object Coriolis terms $C_o(X, \dot{X})$ can be expressed as,

$$C_o(X, \dot{X}) = \begin{bmatrix} 0 & 0 \\ 0 & c_o + g_o + I_{pqr} \end{bmatrix} \in \mathbb{R}^{6 \times 6}, \quad (2.51)$$

where

$$\begin{aligned} c_o &= \begin{bmatrix} \sin \theta \sin \psi & \cos \psi & 0 \\ \sin \theta \cos \psi & -\sin \psi & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{\psi} & g_o &= \begin{bmatrix} -\cos \theta \cos \psi & 0 & 0 \\ \cos \theta \sin \psi & 0 & 0 \\ -\sin \theta & 0 & 0 \end{bmatrix} \dot{\theta} \\ I_{pqr} &= \begin{bmatrix} (I_{zz} - I_{yy})(\psi + \phi \cos \theta)(\dot{\theta} \cos \psi + \dot{\phi} \sin \theta \sin \psi) \\ (I_{xx} - I_{zz})(\psi + \phi \cos \theta)(\dot{\theta} \sin \psi - \dot{\phi} \sin \theta \cos \psi) \\ (I_{yy} - I_{xx})(\dot{\theta} \sin \psi - \dot{\phi} \sin \theta \cos \psi)(\dot{\theta} \cos \psi + \dot{\phi} \sin \theta \sin \psi) \end{bmatrix} \end{aligned} \quad (2.52)$$

The resultant force exerted on the object is $F_o = \begin{bmatrix} F_i^T & M_i^T \end{bmatrix}^T \in \mathbb{R}^6$ which is a vector of the external forces and the moment exerted on the object. It is also called the external wrench acting on the body centre of mass. All the matrices presented in this section will be used in future chapters for control purposes.

2.8. Summary

In this chapter, the mathematical model of a multi-fingered Barrett Hand is developed using the Lagrangian approach. First, the forward and inverse kinematics of each fingers are formulated and the workspace range for the task frame is determined. The dynamics of the hand is presented in a structured format and all matrix elements are presented. Matrix controller properties are discussed and will be used in the subsequent chapters for controller design. The SimMechanics based physical model of the Barrett hand is developed. This physical model will be used for implementing controllers.

The mathematical model of the object is also developed in this chapter. In this case, the Newton-Euler method is used to derive the dynamics. First, the object dynamics is expressed

2.8 Summary

with respect to the velocity coordinates of the object body. Then, it is converted in the local coordinates to have the local parametrization of $X_o \in \mathbb{R}^6$. The Euler $Z - Y - Z$ transformation is used to convert the angular object velocity in order to achieve the orientation angle of the object. Similarly, a SimMechanics model of the object is developed in this chapter. The mathematical models developed here will be used as test bed platform for controller implementation and assess the performance of the Barrett hand.

3. Adaptive and Higher order sliding mode control of the Barrett hand

3.1. Introduction

Robot control areas determine the way how a robot system is developed to perform a desired task efficiently. Control design of a robot system may be divided into few steps: firstly, good understanding of the physical robot system, secondly, the mathematical modelling of the robot and finally the control specifications. To understand the nature and behavior of the physical system, the physical variables whose behaviour is to be controlled must be identified. Depending on the domain type, these physical variables are the joint motion or the Cartesian motion, also called the output of the system. These outputs are measured subject to the input voltage when the joint motor dynamics are considered. Otherwise, force or torque can be assumed arbitrary input for control of the robot system. So the output of the system can be defined as,

$$y(t) = y(q, \dot{q}, f) = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \quad (3.1)$$

According to the equation (3.1), the dynamics are formed which is either analytical or experimental. Analytical dynamics for a robotic hand has been derived in chapter 2 with the advantage of forming a precise model of the system. Experimental procedure requires experimental data collected from the system and observe the system behaviour based on the different input signals. Compared to the analytical model, this method is imprecise as it depends on the system input. Hence, a mathematical model of the robotic hand is used here for control implementation. From chapter 2.4, the robotic hand dynamics can be written as,

$$M_{f_i} \ddot{Q}_{F_i} + C_{f_i}(\dot{q}_i, \dot{\alpha}_i) + G_{f_i} = \tau_{f_i} \quad (3.2)$$

The above robot dynamics is expressed as ordinary nonlinear and non autonomous differential equations. This type of motion equation limits the application of linear control technique in the robotic system such as robotic hand. Furthermore, the tracking accuracy requirements of the robotic hand for grasping and manipulation tasks demands sophisticated control techniques.

Hence, the robot dynamics are suitable with nonlinear and adaptive controllers. The control specification of the robotic hand system needs to be analysed first for developing controllers. The general control objective for any system can be presented in the following steps:

- Stability
- Tracking
- Optimisation

Stability is the most important specifications for any control system. A control system cannot be authenticated without achieving the knowledge of the stability issues. The stability concepts discuss the system properties within designated equilibrium or as closely to it-as possible for an infinite length of time. Two stability techniques are available to discuss the analytical studies of the robot control. The first theorem is called *Lyapunov Stability* theory and the second one is the *input-output* stability theory. The stability theorems are well described by the second method or direct method of Lyapunov. The objective of the Lyapunov stability theory is to analyse the behaviour of dynamical system expressed as ordinary differential equation form below,

$$\dot{x} = f(t, x) \quad x \in \mathbb{R}^n, \quad (3.3)$$

where the vector x is the state of the system. The solution of the above differential equation is denoted by $x(t, t_o, x(t_o))$ which is the evaluation of system's state at time t with initial state $x(t_o) \in \mathbb{R}^n$ and time condition $t_o \geq 0$. Both initial conditions are assumed to be fixed for simplicity. For each initial condition $t_o, x_o(t_o)$, a unique solution is found for (3.3). The solution of system exists for all $0 \leq t_o \leq t$ is bounded and only available on finite interval. This is referred to as non autonomous system where initial time t_o is given to achieve the solution. The robot dynamics system lies in the category of autonomous differential equation which allows to consider the initial time as $t_o = 0$ and its solution is assumed to be available on infinite interval. In that case, the equation (3.3) becomes,

$$\dot{x} = f(x) \quad x \in \mathbb{R}^n, \quad (3.4)$$

which is called the autonomous system.

Tracking is generally a control objective which states that the desired time based input trajectory should be followed by the output motion of the system. Based on 3.1, the system output y corresponds to the joint position q_i in the joint space or the Cartesian position $X \in \mathbb{R}^3$ in the workspace. The joint space and workspace tracking are defined in (3.5) as,

$$\begin{aligned} \lim_{t \rightarrow \infty} q(t) &\rightarrow q_d(t) \\ \lim_{t \rightarrow \infty} X(t) &\rightarrow X_d(t) \end{aligned} \quad (3.5)$$

Tracking can be only maintained in the above two equation when the output $q(t)$, $X(t)$ follows $q_d(t)$, $X_d(t)$. This is the fundamental control problem in robotics to achieve the desired task followed by the joint or the workspace configuration of the robotic system. The desired trajectory can be generated according to the task specification in the joint or the workspace. When the trajectory is generated at the workspace, inverse kinematics is applied for implementing joint based controller.

Optimisation is the last step of the control specification where the control method is discussed followed by the stability and the tracking control problem. The difference between the actual and the desired motions can be defined as e_q and e_X . Now, the (3.5) can be re-written as,

$$\begin{aligned} \lim_{t \rightarrow \infty} e_q &\approx 0 \\ \lim_{t \rightarrow \infty} e_X &\approx 0 \end{aligned} \quad , \quad (3.6)$$

which is the control problem in terms of the system state error. The error should converge to zero over time. This problem can be solved by choosing different control methods. In this case, the objective is to steer the output to the desired state so that the error can be minimised.

3.2. Equilibrium and Lyapunov Stability

To realise the stability and control problem issues discussed in the section 3.1, the concept of system equilibrium is important. A constant vector x_e is said to be in equilibrium or in an equilibrium state if

$$f(t, x_e) = 0 \quad \forall t \geq 0. \quad (3.7)$$

It can be also said that, if the initial state of the system $x(t_o) \in \mathbb{R}^n$ is in equilibrium then the system state and its derivative corresponds to

$$\begin{aligned} x(t) &= x_e & \forall t > 0 \\ \dot{x}(t) &= 0 & \forall t > 0 \end{aligned} \quad . \quad (3.8)$$

For autonomous system, it is assumed that the origin of state space $x = 0$ is an equilibrium of (3.3). Considering this, the stability definition will be presented below,

Definition 3.1: The origin of state space is a stable equilibrium (in the sense of Lyapunov) of (3.3) if, for each $\varepsilon > 0$, there exists $\delta = \delta(\varepsilon) > 0$ such that,

$$\|x(0)\| < \delta \implies \|x(t)\| < \varepsilon \quad \forall t > 0. \quad (3.9)$$

Similarly, the origin of (3.4) is stable if for each $\varepsilon > 0$ there exists $\delta = \delta(\varepsilon) > 0$ such that (3.9) holds with $t_o = 0$. It is apparent from (3.9) that, the constant δ is smaller than ε and is not

unique and also that the term δ depends on ε [86]. The value of δ should be produced such that a trajectory starting in a δ neighbourhood of the origin will never leave neighbourhood of ε . It should be also noted that, $\delta > 0$ applied for each and every $\varepsilon > 0$. The equilibrium $x = 0$ of (3.4) is unstable if it is not stable and called asymptotically stable if it is stable and δ can be selected such that

$$\|x(0)\| < \delta \Rightarrow \lim_{t \rightarrow \infty} x(t) = 0. \quad (3.10)$$

Uniform asymptotic stability is the main term of interest in robotics. For example, a robot manipulator needs to move to a point which means the system state should converge to that point, not to remain in nearby neighbourhood. Uniformity only appears in time varying system [60]. For autonomous system, asymptotic stability implies uniform asymptotic stability though unable to determine the rate of convergence to the origin. Exponential stability indicates the rate of convergence. The equilibrium $x = 0$ of (3.4) is an exponentially stable equilibrium point if there exists constant terms $p, \mu > 0$ and $\varepsilon > 0$ such that

$$\|x(t)\| \leq p e^{-\mu t} \|x(0)\| \quad (3.11)$$

for all $\|x(0)\| < \varepsilon$ and $t > 0$. The constant μ determines the rate of convergence. Exponential stability is secure as it also implies asymptotic stability. This type of stability is important when advanced control algorithm such adaptive control is considered.

Based on the stability definitions presented above, the direct/second method of Lyapunov can be discussed. This method does not require integration of the system (3.4) in order to check the stability. It is based on the energy measurement of the system. Then the rate of change of energy is analysed to determine stability [86]. The term *measurement of energy* can be defined as follows: let B_ε be a ball of size ε around the origin, $B_\varepsilon = \{x \in \mathbb{R}^n : \|x\| < \varepsilon\}$. Now few definitions are required to state,

Definition 3.2: Locally positive definite function

A continuous function $V : \mathbb{R}^+ \times \mathbb{R}^n$ is locally positive definite (l.p.d) if there exists some $\varepsilon > 0$ and few continuous, strictly increasing function $\alpha : \mathbb{R}_+ \rightarrow \mathbb{R}$,

$$V(0, t) = 0 \quad V(x, t) \geq \alpha(\|x\|) \quad \forall x \in B_\varepsilon, \forall t \geq 0. \quad (3.12)$$

Locally positive definite function behave as local energy function.

Definition 3.3: Positive definite function

A continuous function $V : \mathbb{R}^+ \times \mathbb{R}^n$ is positive definite if it satisfies the **Definition 3.2** and also follows $\alpha(p) \rightarrow \infty$ as $p \rightarrow \infty$.

Definition 3.4: Decrescent function

When the energy functions are required to be bounded, decresence function is defined. A

3.2 Equilibrium and Lyapunov Stability

continuous function $V : \mathbb{R}^+ \times \mathbb{R}^n$ is called decrescent if for some $\varepsilon > 0$ and few continuous, strictly increasing function $\beta : \mathbb{R}_+ \rightarrow \mathbb{R}$,

$$V(x, t) \leq \beta(\|x\|) \quad \forall x \in B_\varepsilon, \forall t \geq 0 \quad (3.13)$$

The above three definitions are important to ascertain stability for the system by the Lyapunov theorem with appropriate energy based function [15]. In general, the theorem states that when $V(x, t)$ is a locally positive definite function and $\dot{V}(x, t) \leq 0$ then stability of the equilibrium is found. The time derivative of the energy function V is taken along the system trajectory as,

$$\dot{V} \big|_{\dot{x}=f(x,t)} = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f \quad (3.14)$$

Theorem 3.1: Lyapunov theorem

Consider $V(x, t)$ as a non-negative energy function with time derivative \dot{V} along system trajectories.

1. If $V(x, t)$ is locally positive definite and $\dot{V}(x, t) \leq 0$ in x locally and for all t , then the system origin is locally stable (In the sense of Lyapunov).
2. If $V(x, t)$ is locally positive definite and decrescent, and $\dot{V}(x, t) \leq 0$ locally in x and for all t , then the system origin is uniformly locally stable (In the sense of Lyapunov).
3. If $V(x, t)$ is locally positive definite and decrescent, and $-\dot{V}(x, t)$ is locally positive definite, then the system origin is uniformly locally asymptotically stable.
4. If $V(x, t)$ is positive definite and decrescent, and $-\dot{V}(x, t)$ is positive definite, then the system origin is globally uniformly asymptotically stable.

Theorem 3.1 provides conditions for determining the stability conditions of the system origin [15]. The only drawback is that it does not show how to address the Lyapunov function $V(x, t)$ but it definitely exists as if an equilibrium point is stable. To determine the convergence rate of solutions to the equilibrium, this theorem is limited. Hence, the exponential stability theorem is needed considering energy function.

Theorem 3.1: Exponential stability theorem

The origin $x = 0$ is an exponentially stable equilibrium point of $\dot{x} = f(x, t)$ if and only if there exists an $\varepsilon > 0$ and a function $V(x, t)$ which satisfies,

$$\begin{aligned} \alpha_1 \|x\|^2 &\leq V(x, t) \leq \alpha_2 \|x\|^2 \\ \dot{V} \big|_{\dot{x}=f(x,t)} &\leq -\alpha_3 \|x\|^2 \\ \left\| \frac{\partial V}{\partial x}(x, t) \right\| &\leq \alpha_4 \|x\| \end{aligned} \quad (3.15)$$

for some positive constant $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ and $\|x\| \leq \varepsilon$. It can be shown that

$$m \leq \left(\frac{\alpha_2}{\alpha_1} \right)^{1/2} \quad \alpha \geq \frac{\alpha_3}{2\alpha_2},$$

which are bounded in the equation (3.11). The proofs are given in [60].

3.3. Motion problem formulation

The motion problem of multi-fingered robotic hands is similar to the open chain manipulator. To grasp an object, the robotic hand needs to reach a desired location from its initial position. This desired position is often described as the posture of the object that is achieved just before grasping. This problem is referred to as the motion problem of the multifingered hand in an open space. It is usually formed in workspace as the object contact locations are assumed to be known and these locations are used as desired location to drive the hand finger joints. There are two different ways to achieve the output in workspace. One way is to define the workspace trajectory, calculate the inverse kinematics and define the control problem in the joint space. Then the joint output can be transformed to the workspace with forward kinematics. Or alternatively, the control problem can directly be defined in the workspace. Joint based control is considered here as inverse kinematics of the Barrett hand is easy to solve. However, direct workspace based control is also shown at the end of this section. Rewriting the workspace dynamics of the robot finger from (2.24) allows the motion problem to be defined as,

$$M_q(q)\ddot{q} + C_q(q, \dot{q})\dot{q} + G_q(q) = \tau. \quad (3.16)$$

The formal way of describing the motion problem is to find the control torque τ such that

$$\lim_{t \rightarrow \infty} q(t) = q_d(t), \quad (3.17)$$

where $q \in \mathbb{R}^n$ is referred to the joint position and $q_d \in \mathbb{R}^n$ is the desired fingertip trajectory to the contact location. It is assumed that, the desired trajectory is specified for all time t and is twice differentiable. This is the tracking control objective in the joint coordinate. A controller should achieve this or does not depend on studying the stability of the origin of the closed loop system by the Lyapunov method. For stability analyzing purpose, (3.17) can be re-written as,

$$\lim_{t \rightarrow \infty} \tilde{q}(t) = 0, \quad (3.18)$$

where $\tilde{q}(t) \in \mathbb{R}^n$ is the joint position error vector and defined by,

$$\tilde{q}(t) := q_d(t) - q(t). \quad (3.19)$$

Considering (3.19), the control objective can be achieved when the closed loop position error in terms of system state $\begin{bmatrix} \tilde{q}^T & \dot{\tilde{q}}^T \end{bmatrix} = 0 \in \mathbb{R}^{2n}$ is asymptotically stable. In physical system, fingertip positions X are determined from the joint position q by forward kinematics. If the robotic hand model is known and satisfy the conditions $q(0) = q_d(0), \dot{q}(0) = \dot{q}_d(0)$, then the control torque in (3.16) can be chosen as follows,

$$\tau = M_f(q_d)\ddot{q}_d + C_f(q_d, \dot{q}_d)\dot{q}_d + G_f(q_d). \quad (3.20)$$

As both q and q_d have the same form of differential equation with similar initial condition, $q(t) = q_d(t)$ for all $t \geq 0$ which is known as an open loop control law. An issue with open loop control is that the current output state of the system remain unused for choosing the control law which is not very robust. In reality, it is not possible to know the current robot position exactly. Hence, the initial conditions become $q(0) \neq q_d(0)$ and the open loop law does not optimise the error. This problem introduces the requirement of the feedback into the robotic hand system. This feedback should be used such that hand motion trajectory converges to the desired motion. There are many popular methods available for using feedback in implementing the control law. Among those, PD based control and computed torques are popular for robotic manipulators. These laws can be extended for controlling the robotic hand as control structure is similar when those are applied to the open chain manipulators.

3.4. Computed torque control

Computed torque control (CTC) method falls into the category of feedback based control system and is considered as popular control method for robotic system. The nonlinear robot system is linearised first and then the feedback is applied to drive the error dynamics by this method. This method first gained popularity in modern control system theory [88]. The advantage of the computed torque is that it can be extended to the robust and adaptive controllers due to its ability to compensate robot nonlinear dynamics. It is a combination of classical joint or workspace control with modern design techniques. Differentiating the workspace tracking error equation (3.18) twice illustrates the CTC method as,

$$\begin{aligned} \dot{e}_q &= \dot{\tilde{q}}_f = \dot{q}_d - \dot{q} \\ \ddot{e}_q &= \ddot{\tilde{q}}_f = \ddot{q}_d - \ddot{q} \end{aligned} \quad (3.21)$$

The time based desired velocity \dot{q}_d and the acceleration \ddot{q}_d can be easily calculated from input position trajectory. Now, the dynamics of (3.16) can be solved for the joint acceleration \ddot{q} as,

$$\ddot{q} = M_q^{-1} (\tau - C_q(q, \dot{q})\dot{q} - G_q(q)). \quad (3.22)$$

Now, the torque τ can be chosen to cancel all nonlinearities such as Coriolis and gravity matri-

ces as,

$$\tau = (M_q \ddot{q}_d + C_q(q, \dot{q}) \dot{q} + G_q(q)), \quad (3.23)$$

which yields,

$$\ddot{q} = \ddot{q}_d. \quad (3.24)$$

The equation (3.23) is the feed forward component of the CTC law. This provides the torque required to drive the system state along its nominal path. If the initial conditions of the actual position and velocity corresponds to the desired one, the manipulator should follow the desired trajectory. Error is found when $q(t)$ is not identical to $q_d(t)$ and cause difficulties to validate the control law. Therefore, feedback component is required to adjust the torque for reducing errors in the system trajectory. Adding of auxiliary control input modifies (3.24) as,

$$\ddot{q} = \ddot{q}_d + u, \quad (3.25)$$

where u is the new auxiliary control input. This can be chosen based on the Proportional Derivative (PD) method to solve the tracking problem as,

$$u = -K_v \dot{e}_q - K_p e_q. \quad (3.26)$$

Apply (3.26) into (3.25) gives the internal error dynamics of the total control law as,

$$\ddot{e}_q + K_v \dot{e}_q + K_p e_q = 0. \quad (3.27)$$

The equation (3.27) is a linear differential equation which reduces the error between the desired and the actual trajectories. Due to its linearity, it is not trivial to select the control gain K_p and K_v to make the system state stable and show that the origin is exponentially stable which implies that $e \rightarrow 0$ as $t \rightarrow \infty$. Then the total control law can be written as,

$$\tau = M_q (\ddot{q}_d - K_v \dot{e}_q - K_p e_q) + C_q(q, \dot{q}) \dot{q} + G_q(q). \quad (3.28)$$

3.4.1. Proof of computed torque control stability

If the control gains K_p and K_v are positive definite and symmetric matrices, then the control law of (3.28) applied to the hand workspace dynamics of (3.16) provides exponential trajectory tracking.

Proof: The error dynamics of (3.27) can be written in the state space as,

$$\frac{d}{dt} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix}}_A \begin{bmatrix} e \\ \dot{e} \end{bmatrix}. \quad (3.29)$$

Each of the eigenvalues of A matrix in (3.29) has negative real part. Consider $\lambda \in \mathbb{C}$ as an eigenvalue of A which corresponds to eigenvector $v = v_1, v_2 \in \mathbb{C}^{2n}$ and $v \neq 0$. Then,

$$\lambda \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}. \quad (3.30)$$

From (3.30), if $\lambda = 0$ then $v = 0$ and therefore $\lambda = 0$ is not an eigenvalue of A . Now if $\lambda \neq 0$ then v_1 becomes zero when $v_2 = 0$. Hence, it is shown $v_1, v_2 \neq 0$ which implies without loss of generality that $\|v_1\| = 1$. Using this, it can be expressed that,

$$\begin{aligned} \lambda^2 &= v_1^* \lambda^2 v_1 = v_1^* \lambda v_2 \\ &= v_1^* (-K_p v_1 - K_v v_2) \\ &= -v_1^* K_p v_1 - \lambda v_1^* K_v v_1, \end{aligned}$$

where $*$ is the transpose of complex conjugate. Define $\beta = v_1^* K_p v_1$ and $\alpha = v_1^* K_v v_1$ yields the following,

$$\lambda^2 + \alpha\lambda + \beta = 0 \quad \alpha, \beta > 0 \quad (3.31)$$

which means the real part of λ is negative. Therefore, the controller is stable.

The advantage of applying CTC law is that it linearises a nonlinear system which allows the usage of any linear control theories for controller design. It is called feedback linearisation technique.

3.4.2. Computed torque control design for the Barrett hand

The mathematical and physical model of the Barrett Hand was developed in chapter 2. The mathematical model is used to design the controller and applied to the physical SimMechanics system. The CTC structure applied to the Barrett Hand system is presented in Figure 3.1. It is assumed that the workspace input is arbitrary to apply. The trajectory of the workspace is chosen based on the Barrett hand workspace given in chapter 2. Then inverse kinematics solution of the Barrett Hand is used to convert the workspace trajectory in the joint space (inverse kinematics subsystem) and the controller is designed in the joint space. The SimMechanics Hand model can provide joint outputs which are fed back to the controller for error measurement. Fingertip outputs are also measured from the hand and are used to verify the tracking based on workspace input. Feed forward block cancels the nonlinear terms and feedback block calculates the error dynamics as auxiliary control. The outer joints are added with the damper to balance the link l_3 of all fingers. The damping structure is shown in Figure 3.2 for finger F_3 . The velocity is measured from the output channel 2 and multiplied by the damping gain of 0.5 and added with control torque as additional input to the SimMechanics based Barrett hand model. This helps to balance the outer link connected to the outer joint for all fingers.

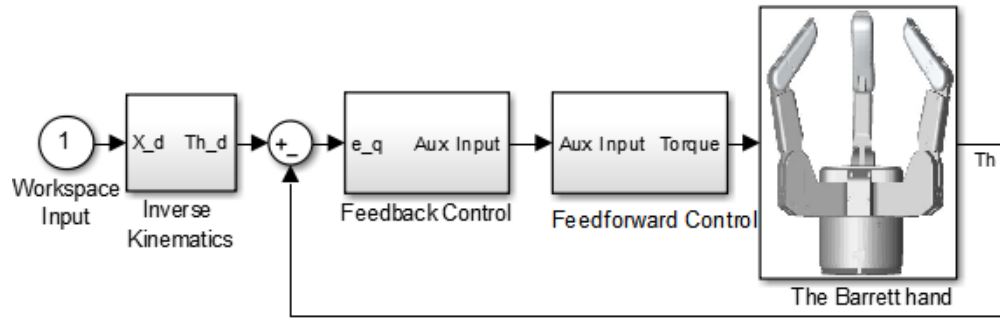


Figure 3.1.: CTC Simulink based controller for the Barrett hand simulation.

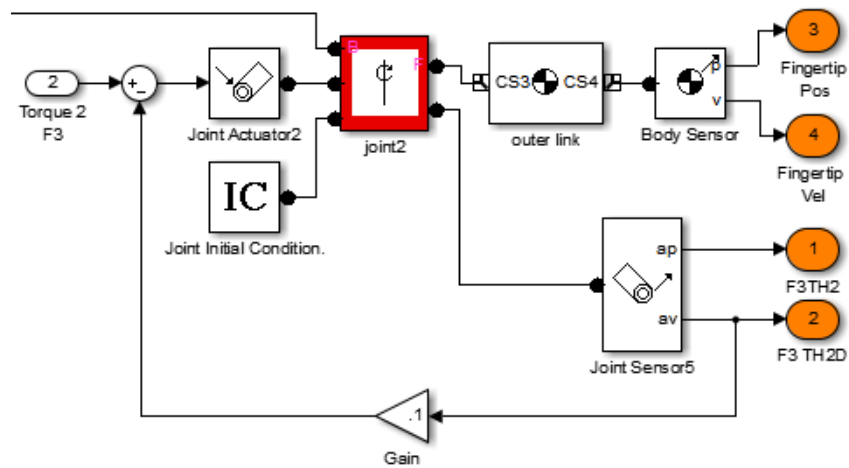


Figure 3.2.: The damping added to the outer joint for balancing outer link (Finger F_3).

3.4.3. Results and discussions

Different workspace trajectories are chosen to see the CTC controller performance. Initial conditions for the workspace and joint space coordinates are given in Table 3.1.

Table 3.1.: Initial joint and workspace positions of the Barrett Hand.

Finger F_i	Initial fingertip position F_{fi} (m)	Initial joint position q_2, q_3 and α (m)
F_1	(.0250, -0.1568, 0.1122)	($2.46^\circ, 40^\circ, 0$)
F_2	(-0.0250, -0.1568, 0.1122)	($2.46^\circ, 40^\circ, 0$)
F_3	(0.00049, 0.1568, 0.1122)	($2.46^\circ, 40^\circ, 0$)

The desired trajectories to be tracked by the Barrett finger joints are applied in three sets. The first set is considered without the spread angle α . The results of the fingertip tracking are shown in Figure 3.3 whereas the joint motion tracking of the fingers are plotted in Figure 3.4. The control gains K_p and K_v in (3.26) are chosen as 400 and 160 respectively based on the trial. The purpose of the control gain tuning was to reduce the overshoot, improve rise and settling time of the joint transient response. In Figure 3.3, the fingertip motions are well tracked for a chosen trajectory. In horizontal direction of the fingertips (see first column of Figure 3.3) for all three fingers, no overshoots are visible but tracking errors are large before settling down to the desired value around 2 seconds. The rise times (90% of the steady state response) of the position responses are found less than 2 seconds in each cases. Considering the settling and rise time, the actual position responses are quick in terms of desired input. The maximum tracking errors are found 5% of the steady state values. The errors are occurred due to the unknown dynamics terms (joint frictions, frictions due to the internal mechanism between finger links) which is not dealt by CTC controllers. Overshoots are found in vertical direction of all (Second column of Figure 3.3) fingertips due to the damping behaviour of the distal joints (q_3) of the fingers. The damping behavior of q_3 is understood by the serial connection between q_2 and q_3 . The joint q_3 is not directly connected to the base; linked to the joint q_2 which causes damping at the joint q_3 . The joint performances are shown in Figure 3.4. Desired Joint inputs are given as $q_2 = 45^\circ$ and $q_3 = 55^\circ$ to assess the joint tracking performances. In Figure 3.4, left column sub figures represent the tracking motion of q_2 . No large tracking errors are visible in this case and average settling time is 2 seconds for all fingers. The motion of q_3 are plotted in the right column sub figures. In this case, tracking errors are present for finger F_1 and F_2 until the actual joint responses reached the final values of 55° . The errors are occurred due to damping of the joint q_3 for all fingers but converged to the desired motion after short time.

The velocities of the joints and the joint control torques of all fingers are shown in Figure 3.5 and Figure 3.6 respectively. In Figure 3.5, the velocities of all fingers are exponentially converged to zero which implies that the system energies are dissipated along with simulation time. In Figure 3.6, the computed torques τ_i ($i = 1, 2$) for all joints are shown. The second joints responses of all fingers show a little oscillation between 0-1 seconds but become linear

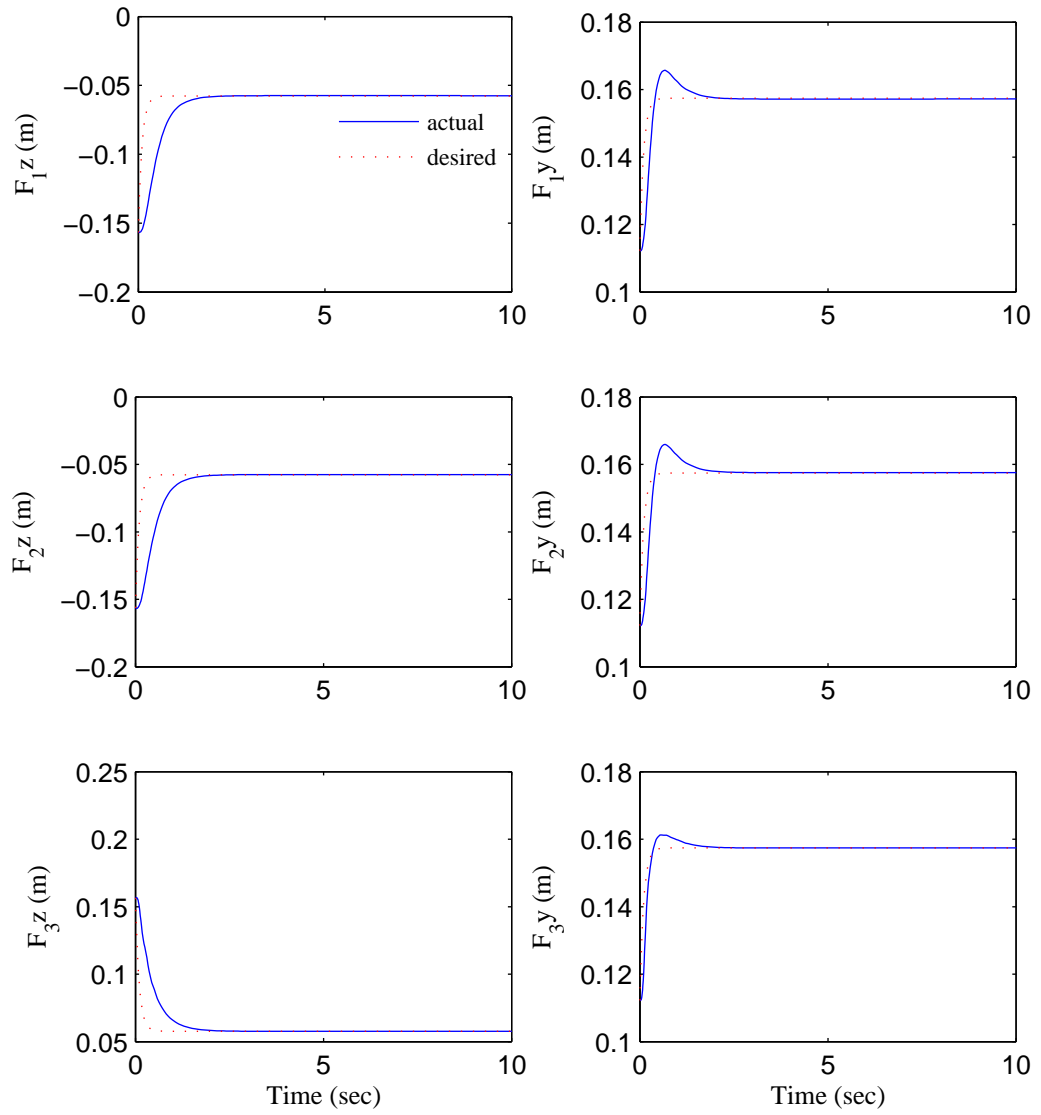


Figure 3.3.: Actual fingertip position tracking of the Barrett fingers for a desired Cartesian trajectory.

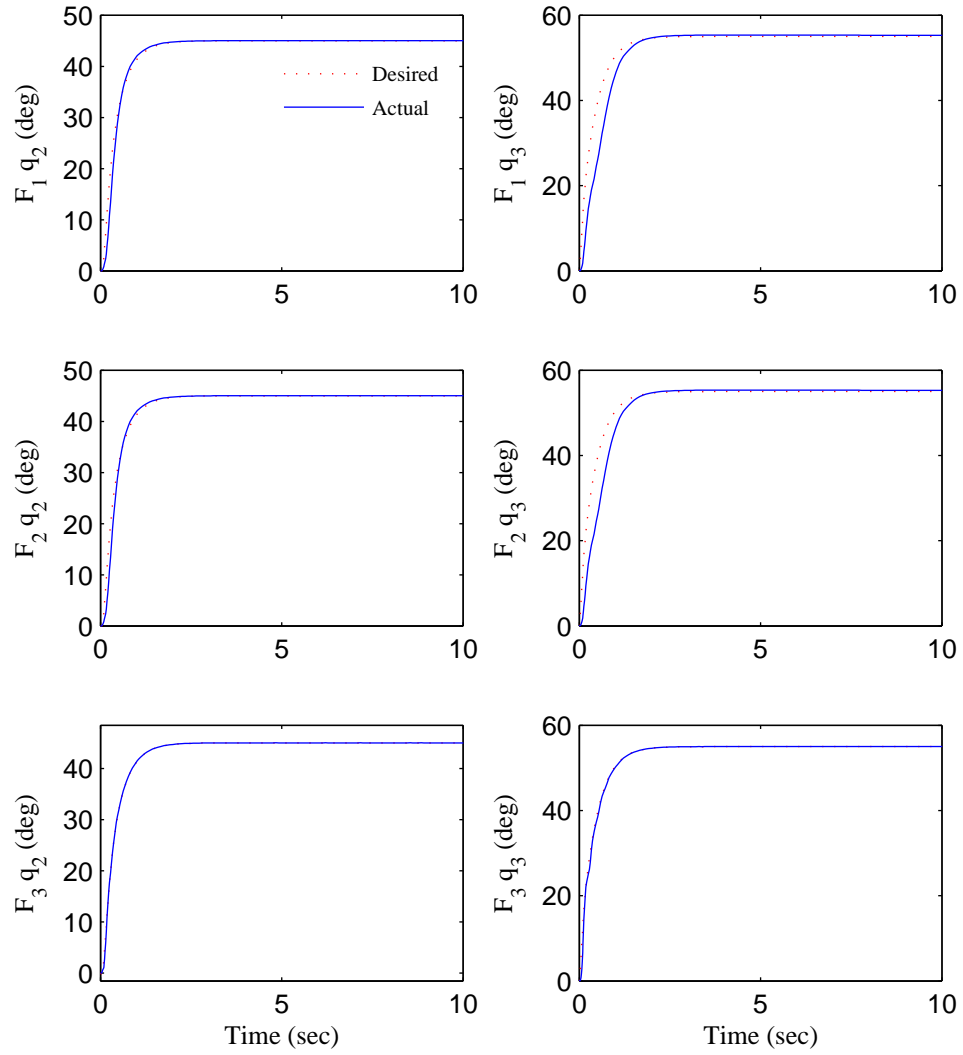


Figure 3.4.: Joint actual (blue) position tracking of the Barrett fingers for the joint position $q_2 = 45^\circ$ and $q_3 = 55^\circ$.

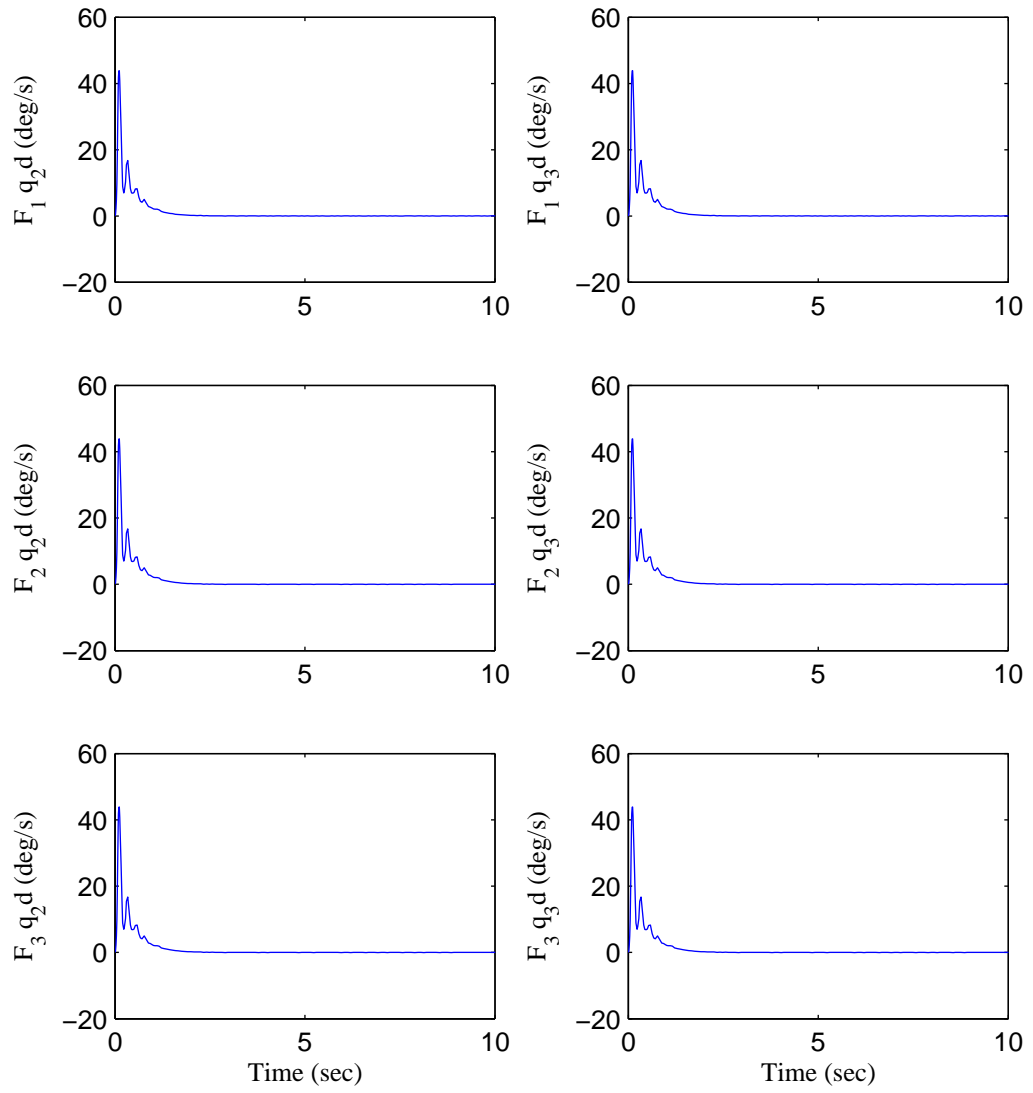


Figure 3.5.: Joint velocities (\dot{q}_i) of Barrett fingers for the joint position, $q_2 = 45^\circ$ and $q_3 = 55^\circ$.

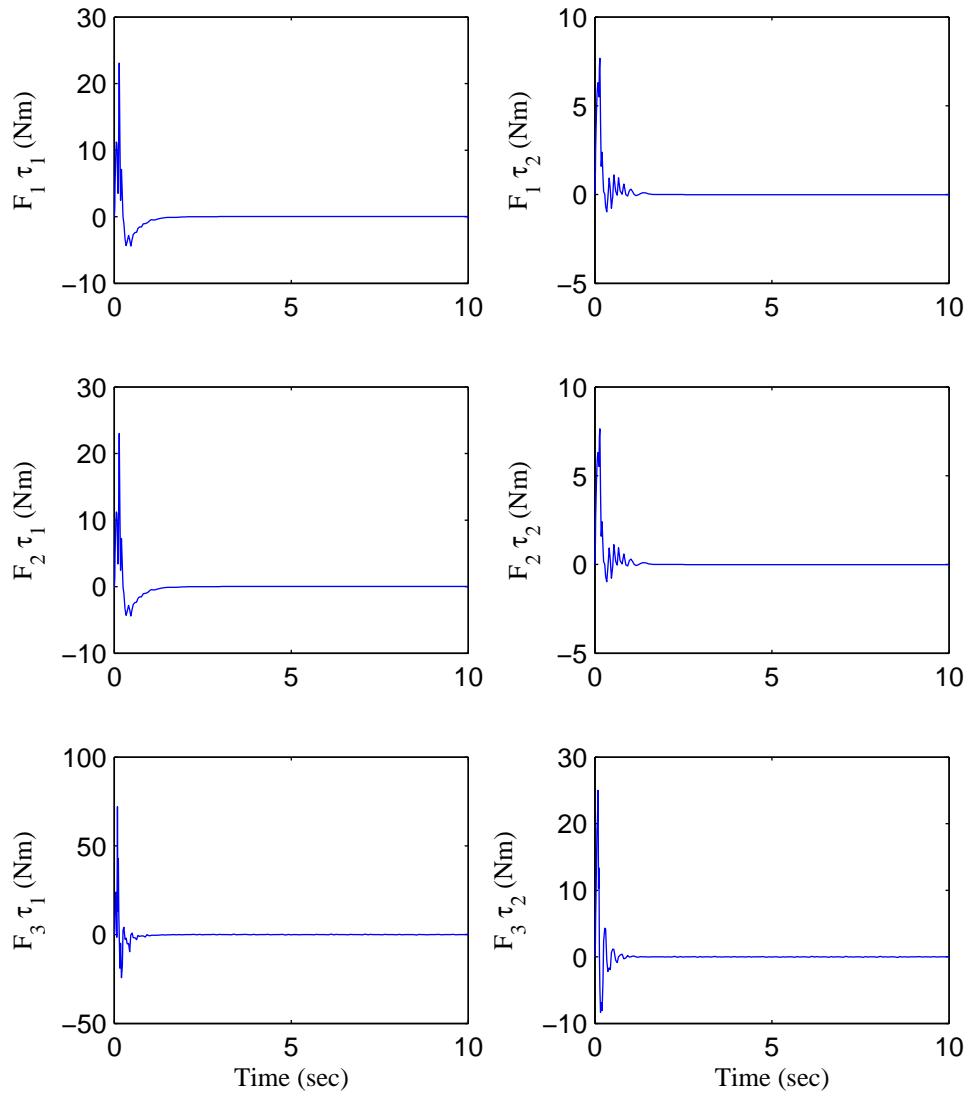


Figure 3.6.: Joint torque (τ) of the Barrett fingers for the joint position, $q_2 = 45^\circ$ and $q_3 = 55^\circ$.

after that. This indicates the smoother torque computed by the control law of (3.28) which drives the joints of each fingers without any oscillation or large overshoot.

The joint position tracking of the fingers are shown in Figure 3.7 for Simulink based Barrett hand model. The desired spread angle is chosen as $\alpha_{di} = 65^\circ (i = 1, 2)$. When the spread angles sp_1 and sp_2 reached 65° , all revolute joints of three fingers started moving and tracked desired motion of $q_{d2} = 45^\circ$ and $q_{d3} = 55^\circ$. In sub figures of the left column, when joint angle sp_1 reaches the desired angle of 65° , the joint position q_2 of all fingers (F_1, F_2, F_3) starts tracking the joint position. For the angle sp_1 , the settling time is less than 2 sec. The joint position of q_2 for the finger F_3 has little overshoot (6% of the final value $q_{di} = 45^\circ$) whereas fingers F_1 and F_2 have no overshoots. The settling time of joint q_{2,F_3} is slow due to the overshoot compared to the joints of q_{2,F_1} and q_{3,F_1} . It is visible from the right column figures that, joint angle sp_2 also achieves the reference values of 65° within settling time of 2 seconds. But the joint position q_3 of all fingers have overdamped response with delayed settling time compared to the q_2 joints. It is depicted from the left and right figure columns that joint responses of q_2 have quick settling time (~ 1.5 seconds) compared to the time found (~ 2 seconds) with q_3 .

In Figure 3.8, the SimMechanics based model is also generated to see the tracking performance of the CTC based controller. In this model, robotic hand dynamics are simulated with differential equations of each fingers. The settling times of spread joints sp_1 and sp_2 are 5 seconds approximately. The execution time is long in SimMechanics based model as it presents each component of the Barrett hand considering inertia, joints frictions and damping. The absence of overshoot, underdamped or overdamped responses are seen in joints q_2 and q_3 of all fingers. No large tracking errors are visible which indicates that the CTC based controller is successful in canceling nonlinear system dynamics and actual responses followed the desired trajectory without oscillations.

The joint tracking errors of the SimMechanics and Simulink hand model are presented and compared in the Figure 3.9 and 3.10. The execution time (2.4 seconds) in SimMechanics appeared to be slower than the Simulink (1.4 seconds) model. It is expected because each component of the physical model has more internal mechanisms which requires more time to execute and exhibits a similar behaviour of the real system. The spread joint reach the desired position within 1 second in Simulink where it takes more than 1.5 seconds in SimMechanics. In both cases, average settling time is faster in simulink ($\sim 1 - 1.5$ seconds) than SimMechanics ($\sim 2.5 - 3.5$ seconds). The tracking errors move towards very close to zero exponentially. The results from Figure 3.3-3.10 show the tracking ability of the CTC controller for known dynamics of the the Barrett hand for any desired trajectory.

In this section, the tracking control problem of a multi-fingered hand is defined. The stabilities of a system are discussed with regards to the tracking control problem. The computed torque method is proposed and implemented to the Barrett Hand for joint and fingertip position tracking. The stability of the proposed controller is proved based on the stability theories discussed

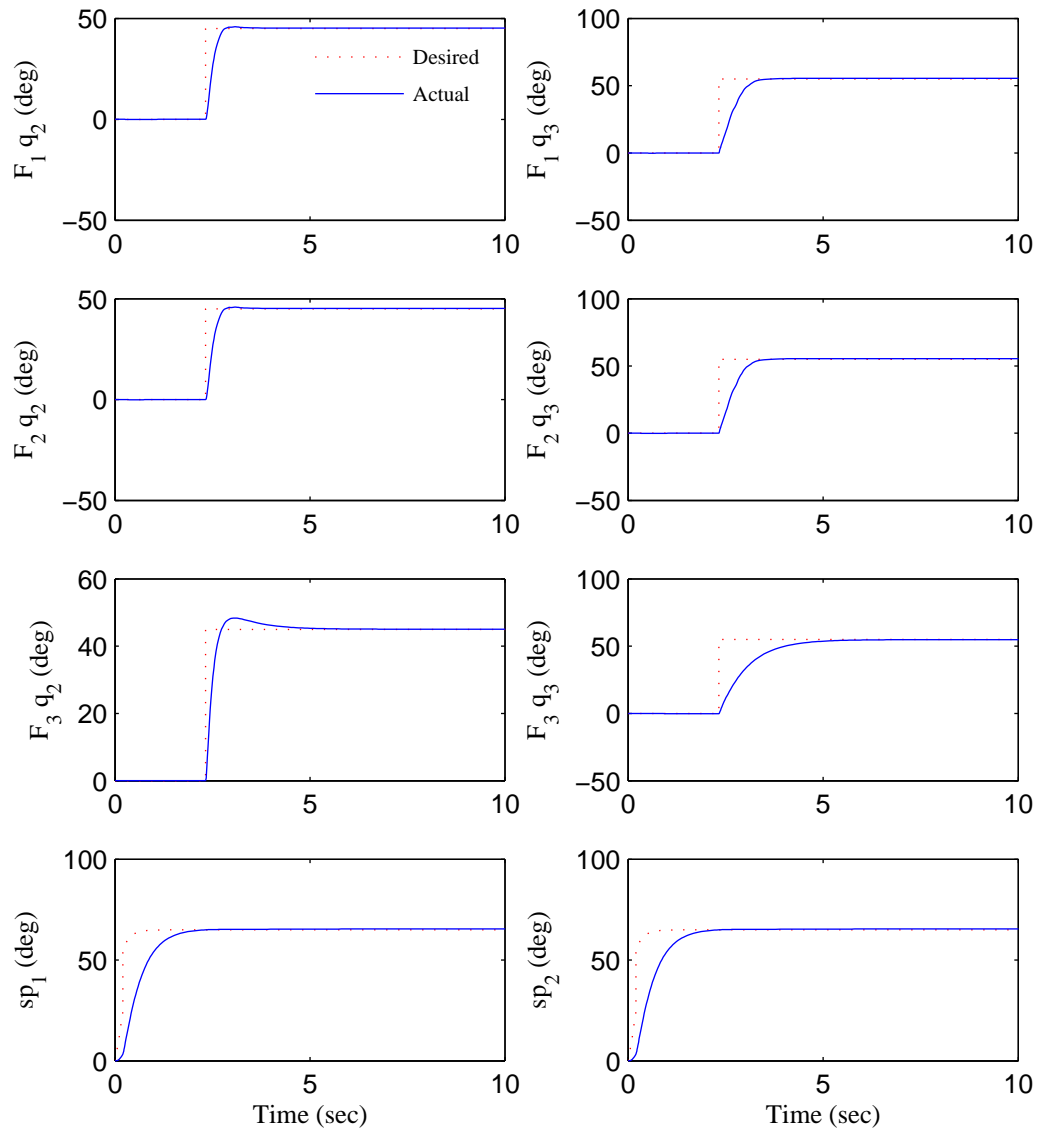


Figure 3.7.: Joint space tracking of Barrett fingers in Simulink for the spread position, $\alpha = 65^\circ$, joint position, $q_2 = 45^\circ$ and $q_3 = 55^\circ$.

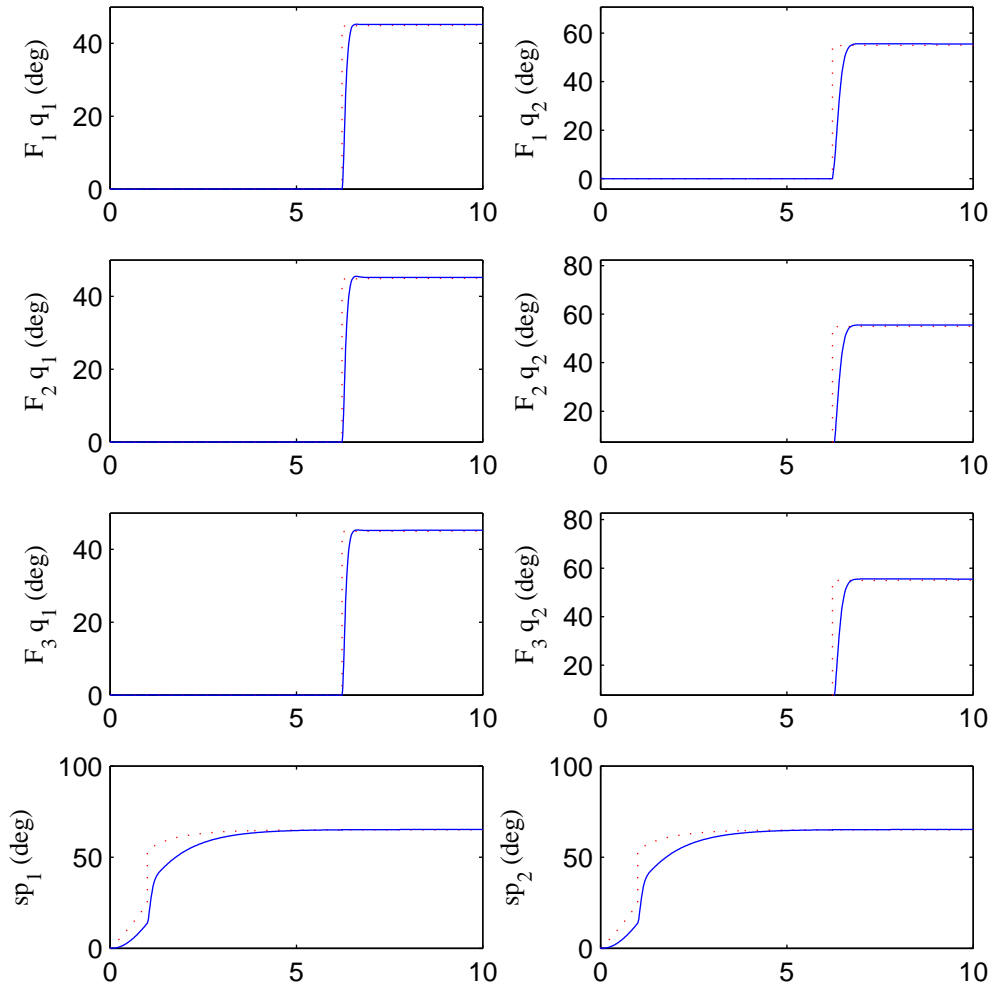


Figure 3.8.: Joint space tracking of the Barrett fingers in SimMechanics for the spread position, $\alpha = 65^\circ$, joint position, $q_2 = 45^\circ$ and $q_3 = 55^\circ$.

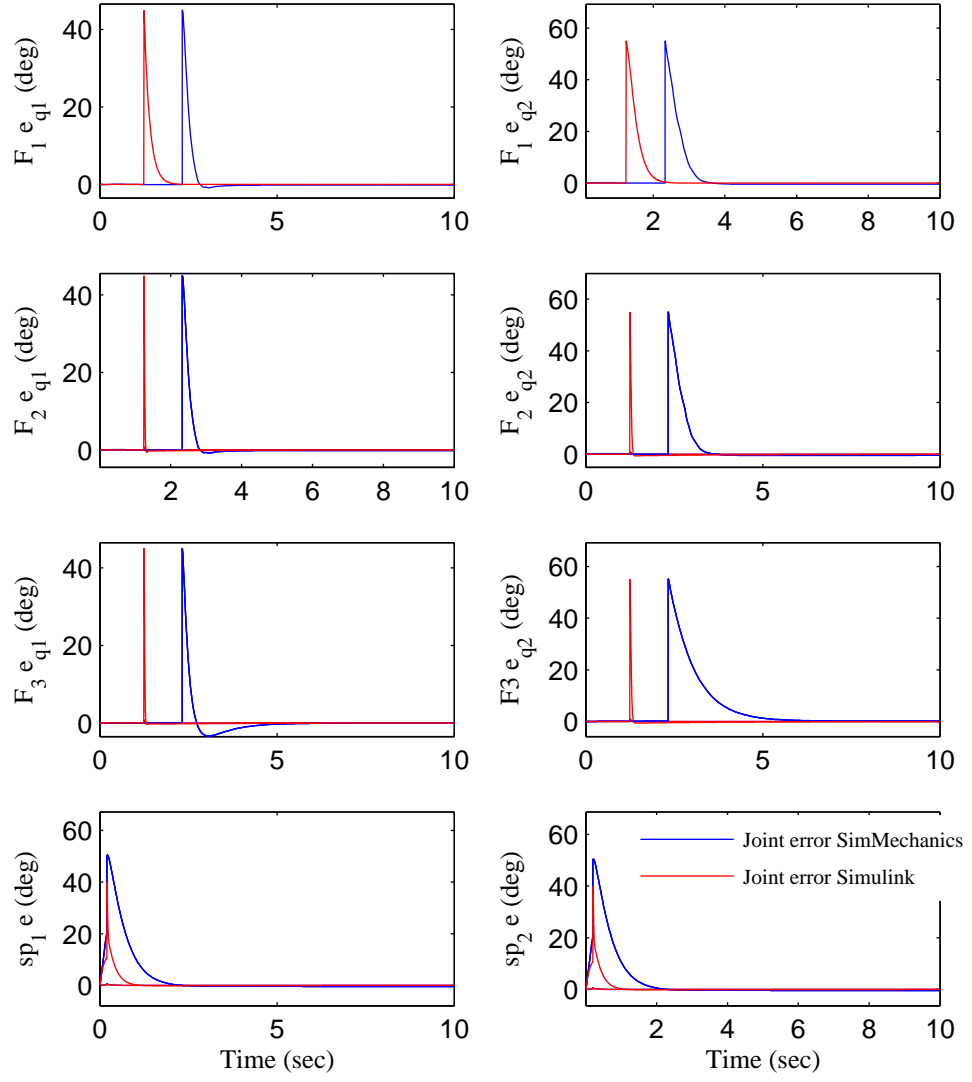


Figure 3.9.: Comparison of the joint tracking error in SimMechanics and Simulink for the spread position, $\alpha = 65^\circ$, joint position, $q_2 = 25^\circ$ and $q_3 = 35^\circ$.

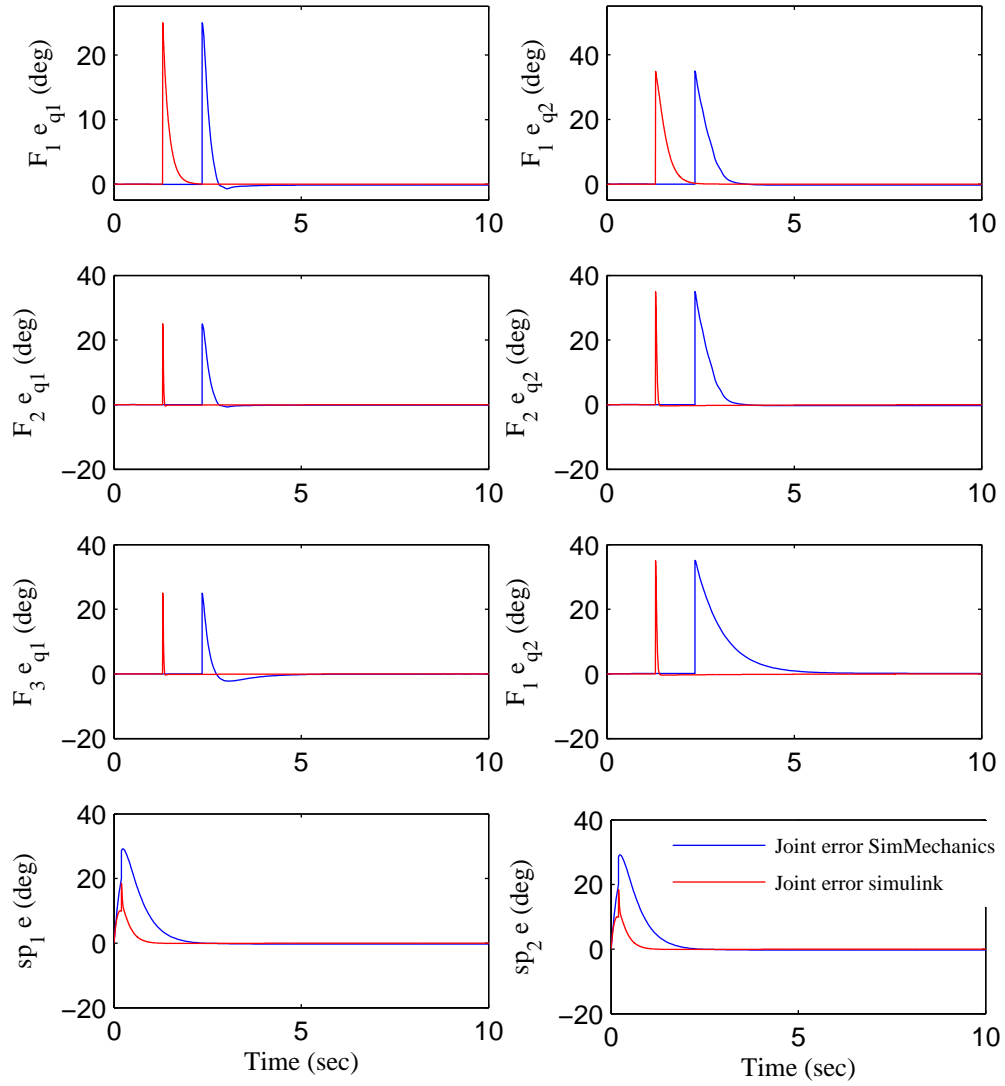


Figure 3.10.: Comparison of the joint tracking error in SimMechanics and Simulink for the spread position, $\alpha = 45^\circ$, joint position, $q_2 = 25^\circ$ and $q_3 = 35^\circ$.

earlier in this chapter. The controller is applied to two model: (1) the SimMechanics and (2) the Simulink model of the Barrett Hand. In both cases, the controller performances are found to be satisfactory but considered under known dynamic parameters. Also, the simulation time of the SimMechanics model is longer than the Simulink model due to its internal complexities rather than just the mathematical equations.

3.5. Sliding and adaptive sliding mode control

The discrepancy between the actual plant and the mathematical model should always be less to design a controller. The discrepancies appear due to reasons such as unknown dynamic parameters and external disturbances. The control law implementation considering any unknown disturbances is always challenging. Another fact is that when less assumptions are considered, the requirement becomes complex in implementing the controller. The same is expected for a multi-fingered robotic hand which is a nonlinear system and contain complex internal mechanism. If these mechanisms are considered as an unknown function or disturbances, CTC based law is unable to compensate the error by feedback based law. Furthermore, the load carried out by the robotic hand makes great change in the performance of the robot workspace. The presence of uncertainties and payload carried by the robotic hand need a robust type controller in order to reduce discrepancies or improve the performance.

Two robust control methods are presented in this section. The first method is called Sliding Mode Control (SMC) and the second one is the Adaptive Sliding Mode Control (ASMC) [89, 90, 86]. Theories for both methods are applied for controller implementation of the Barrett hand. SMC method is capable of dealing with external uncertainties and disturbances where the ASMC shows enhanced performances over varying payload taken by the robotic hand. The idea of SMC is to steer the system state to defined sliding surface so that the unmodelled dynamics and external disturbances can be eliminated. Adaptation is used to estimate the unknown parameters of the system.

3.5.1. Sliding mode theory

The SMC is discussed with respect to the hand dynamics and the joint tracking errors. Consider the robot hand dynamics from (2.15) as,

$$M_f \ddot{q} + C_f(q, \dot{q}) + G_f(q) = \tau_f \quad (3.32)$$

and the joint position tracking errors of the hand is given below,

$$\tilde{q} = q(t) - q_d(t). \quad (3.33)$$

3.5 Sliding and adaptive sliding mode control

A time varying surface $s(t)$ in the state space R^n can be defined by the scalar equation $s(q, t) = 0$ where,

$$s(q, t) = \left(\frac{d}{dt} + \lambda \right)^{n-1} \tilde{q} \quad (3.34)$$

and λ_f is a strictly positive constant. For a second order hand system, the sliding function is defined as,

$$s = \tilde{q} + \lambda_f \tilde{\dot{q}}, \quad (3.35)$$

which implies that the sliding error is nothing but a weighted sum of the position and velocity error [86]. To achieve the tracking using a finite control u (for finite time convergence), the initial condition should satisfy,

$$q_d(0) = q(0). \quad (3.36)$$

The tracking objective of (3.33) with initial conditions of (3.36) is maintained by the sliding surface $s(t)$ for all $t > 0$. The sliding error $s \equiv 0$ provides linear differential equation whose solution is $\tilde{q} \equiv 0$ by initial conditions of (3.36). More simply, the sliding surface $s(t)$ force \tilde{q} to approach zero. The control law τ is chosen to ensure $s(t)$ at the sliding surface following the condition below,

$$\frac{1}{2} \frac{d}{dt} (s^2) \leq -\eta |s|, \quad (3.37)$$

which implies that the distance to the sliding surface decreases along system trajectories and the state trajectories are driven towards the sliding surface where the sliding function is chosen. Before implementing the SMC, virtual reference trajectories are introduced based on the tracking error \tilde{q} to guarantee the convergence of the tracking error. These trajectories are defined as,

$$\begin{aligned} \dot{q}_r &= \dot{q} - s = \dot{q}_d - \lambda \tilde{q} \\ \ddot{q}_r &= \ddot{q} - \dot{s} = \ddot{q}_d - \lambda \dot{\tilde{q}}, \end{aligned} \quad (3.38)$$

where \dot{q}_r is the reference velocity and \ddot{q}_r is the reference acceleration. The term λ is a positive constant. Equation (3.38) shows that the reference velocity \dot{q}_r only increases if the actual output q lags behind the desired value q_d . The same applies to the reference acceleration \ddot{q}_r . Considering the reference input, the SMC based control law is proposed as,

$$\begin{aligned} \tau &= \hat{\tau} - K \text{sgn}(s) \\ \hat{\tau} &= \hat{M}_f \ddot{q}_r + \hat{C}_f \dot{q}_r + \hat{G}_f - K_s s. \end{aligned} \quad (3.39)$$

The SMC law of (3.39) is composed of two parts. The first segment $\hat{\tau}$ includes the feedback linearisation based on reference trajectory and nominal matrices $(\hat{M}_f, \hat{C}_f, \hat{G}_f)$ of the hand dynamics. The second term is taken as **signum** of the sliding function $s(t)$. The control gain K is a positive constant and K_s is also a diagonal positive definite matrix. The stabilities of the SMC controller is proved in **Appendix A.1**

3.5.2. SMC Implementation on planar finger

The SMC controller (3.39) is applied to two link planar finger. The control block diagram of the closed loop system with the SMC controller is given in Figure 4.1.

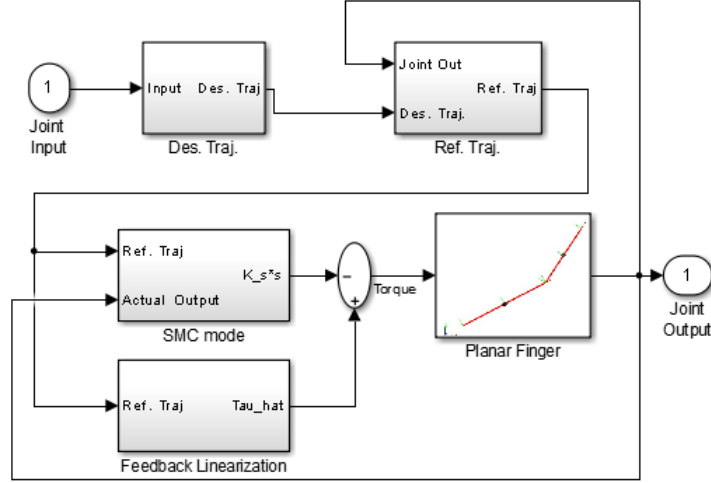


Figure 3.11.: SMC based control structure applied to a planar finger.

It is seen from Figure 3.11 that the reference trajectory is formed from the desired trajectory block and applied to SMC mode block to generate sliding function $s(t)$. It is also applied to calculate linearisation based torque $\hat{\tau}$. Both torques are added to calculate the final joint torque τ and applied to the planar finger.

3.5.3. Simulation and results of sliding mode control

For a planar finger, the same mass and link dimensions of the Barrett finger are used. The initial conditions are changed because of the input structure. A sinusoidal shape input is used to generate trajectories in this simulation. The sliding gain K_{as} and the reference gain λ_{as} are chosen as 11 and 0.03 respectively by trial and error. In Figure 3.12, the joint tracking and velocities, control torque and sliding function results are available.

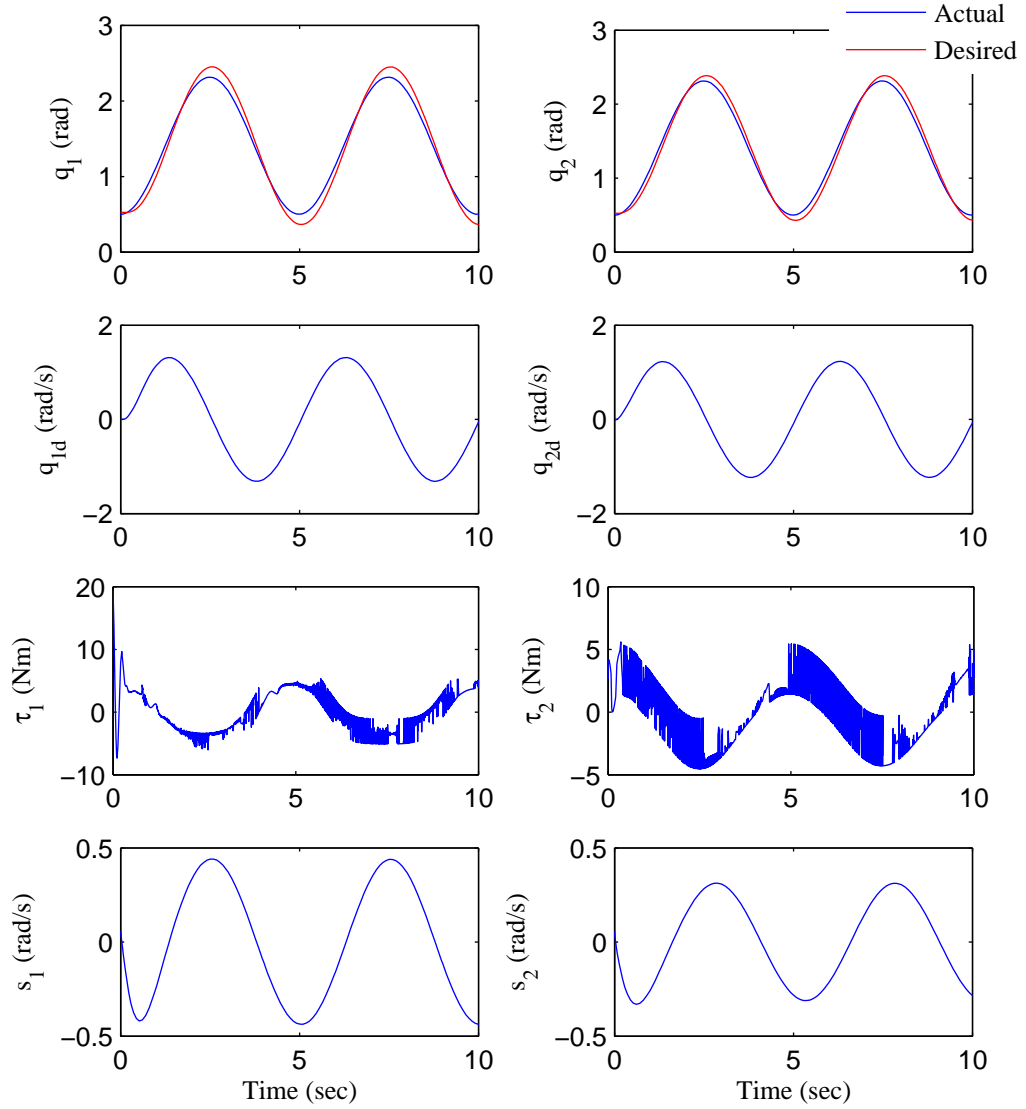


Figure 3.12.: Joint position, velocities, control torque and sliding function results with the SMC controller applied to a planar finger.

The joint motions showed good tracking though the steady state errors between the actual (blue) and the desired (red) motions are clearly visible for joint q_1 and q_2 . The velocities and torque inputs are also shown. The control torque signals are visible with high chattering values. The system was simulated with nominal matrices of the dynamics. The SMC law showed an inability to compensate the varying dynamic parameters to eliminate the tracking errors. An adaptation is required to achieve precise tracking of joint motions.

3.5.4. Adaptive SMC theory (ASMC)

As stated at the beginning of this chapter, the adaptive control is used in robotics for manipulation tasks such as carrying an unknown payload. The unknown payload greatly affects the system performance where the state of the system cannot be compensated for the traditional feedback based law. These payload parameters need to be gradually reduced online by an adaptation or estimation mechanism. Otherwise, it may cause instability for the system. A continuous redesigning of the controller is required as initially chosen parameters may not be suited for the varying system estates. Adaptive control method is effective in estimating the system parameters online which can be used to estimate the system dynamics online and then update the control law [60],[88].

An adaptive controller is different from the conventional controller in terms of the control parameters. These parameters are not fixed and different adjusting mechanisms are used to update these parameters when the system is running. Consider the adaptive controller design problem subject to a dynamical robot system. For a desired robot joint trajectory $q_d(t)$, the tracking problem is to achieve the actual joint position, $q(t) \approx q_d(t)$. In practical situation, it is not possible to know all dynamic parameters of a system accurately due to unknown frictions (joint and tendon) and other internal mechanism. The idea of designing the adaptive controller is to compensate for these parameters. The controller is derived as input torque signal to drive the finger joints and an estimation law is designed for estimating unknown parameters such that $q_d(t)$ is tracked by $q(t)$ after initial adaptation.

Let σ be the unknown parameter vector which contains equivalence of dynamic parameters. The time varying estimation of the parameter vector is $\hat{\sigma}$. Now rewrite the linearly parametrised dynamics of (2.28) with estimation vector σ as follows,

$$\hat{M}_f \ddot{q}_r + \hat{C}_f(q, \dot{q}) \dot{q}_r + \hat{G}_f(q) = Y_f(q, \dot{q}, \ddot{q}_r) \sigma, \quad (3.40)$$

where $Y_f \in \mathbb{R}^{2 \times 5}$ is the parameter free hand dynamics. It is also called regressor matrix and presented in matrix form below,

$$Y_f = \begin{bmatrix} \ddot{q}_{r1} + \ddot{q}_{r2} & \cos(2\ddot{q}_{r1} + \ddot{q}_{r2}) - s_2(\dot{q}_2 \dot{q}_{r2}) - s_2(\dot{q}_{r2} \dot{q}_1) & \ddot{q}_{r1} & gc_1 & gc_{12} \\ \ddot{q}_{r1} + \ddot{q}_{r2} & \ddot{q}_{r1} c_2 \dot{q}_{r1} \dot{q}_1 s_2 & 0 & 0 & c_{12} \end{bmatrix}. \quad (3.41)$$

The terms \dot{q}_{ri} and \ddot{q}_{ri} are the same reference trajectories from the previous subsection which are introduced to converge the tracking error. Now the control law is chosen as,

$$\tau = Y_f(q, \dot{q}, \ddot{q}_r) \sigma - K_{as} s \quad (3.42)$$

3.5 Sliding and adaptive sliding mode control

and the update law for $\sigma \in \mathbb{R}^{5 \times 1}$ is chosen as,

$$\dot{\sigma} = \Gamma_{as}^{-1} Y^T s, \quad (3.43)$$

where (3.42) is the sliding based control with the adaptation mechanism. The sliding and adaptive control gain are defined as K_{as} and Γ_{as} respectively. The control structure of the adaptive sliding mechanism applied to a planar finger is given in Figure 3.13.

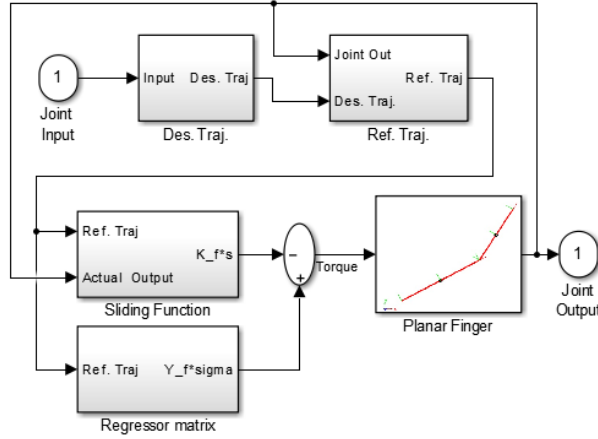


Figure 3.13.: Adaptive-SMC based control structure applied to a planar finger.

In Figure 3.13, the joint angle q and the velocity \dot{q} are used to form the reference velocity \dot{q}_r . The sliding function s and the dynamic regressor matrix Y_f are generated by the reference velocity \dot{q}_r and acceleration \ddot{q}_r . Then, the control input τ is calculated from (3.42) which is applied to the finger. The similar control system is implemented and applied to the SimMechanics based Barrett hand model.

3.5.5. Simulation and results of ASMC

The additional change made to the SMC based planar control system presented in Figure 3.13 is the adaptive update mechanism; the sliding and reference gain remain same. The adaptation gain $\Gamma_{as} = 0.1$ is obtained by trial and error. In Figure 3.14, the trajectories of joint motion q_i are well tracked and oscillation free for a given sinusoidal input. The joint velocities \dot{q}_i appear to be linear. Regarding torque τ , the second joint q_3 required more torque in order to move the outer link. A similar effect is visible for the sliding function s_i which implies that the second link requires more energy to be driven towards the sliding surface. The results of the parameter estimation are given in Figure 3.15. The initial adaptation value $\sigma(0) = 1$ is selected to start the adaptive update process. The reason to choose $\sigma(0) = 1$ is seen from (3.40). The right side of (3.40) is the linear parametrization of the robot dynamics with $\sigma(0) = 1$ is just equal to the

left hand dynamic terms. The update process starts with the known dynamics and estimation process manipulates the right side of (3.40) based on varying payload.

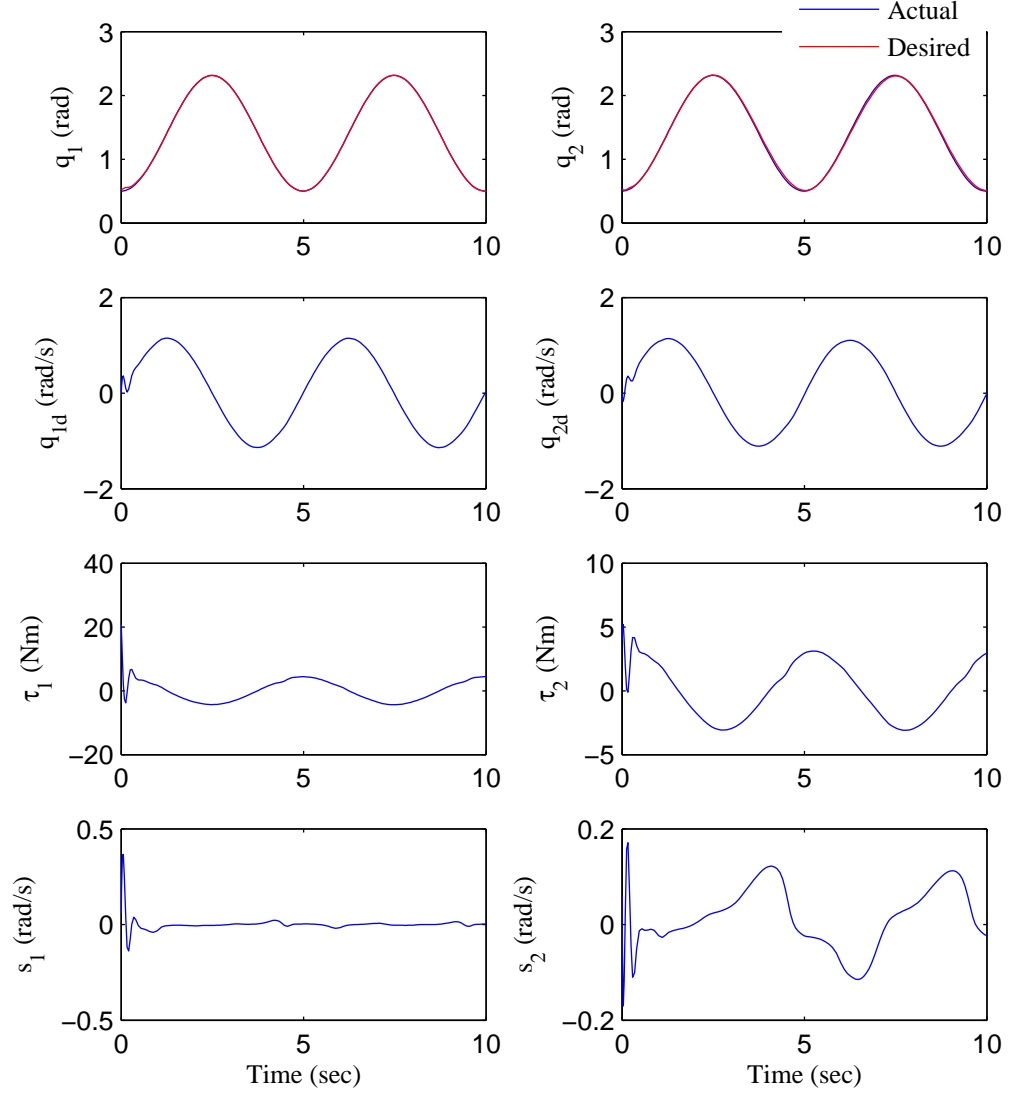


Figure 3.14.: Adaptive SMC results for joint motion q , velocity \dot{q} , torque τ and sliding function s .

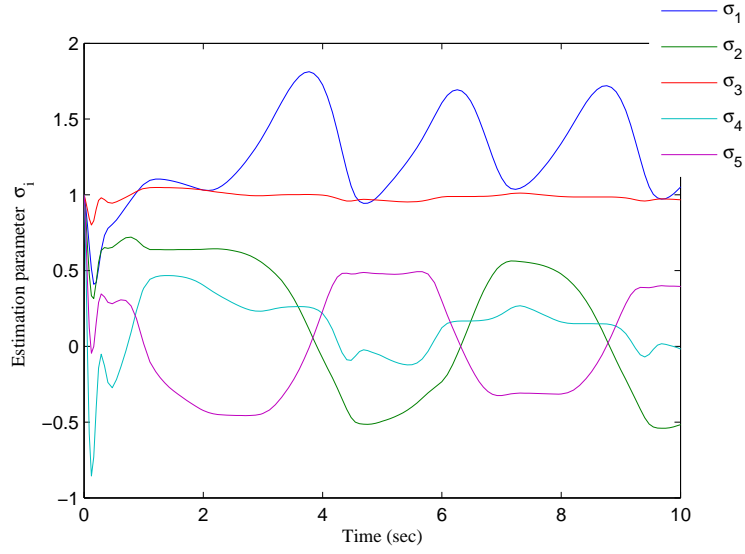


Figure 3.15.: Parameter estimation of σ_i with adaptive update law.

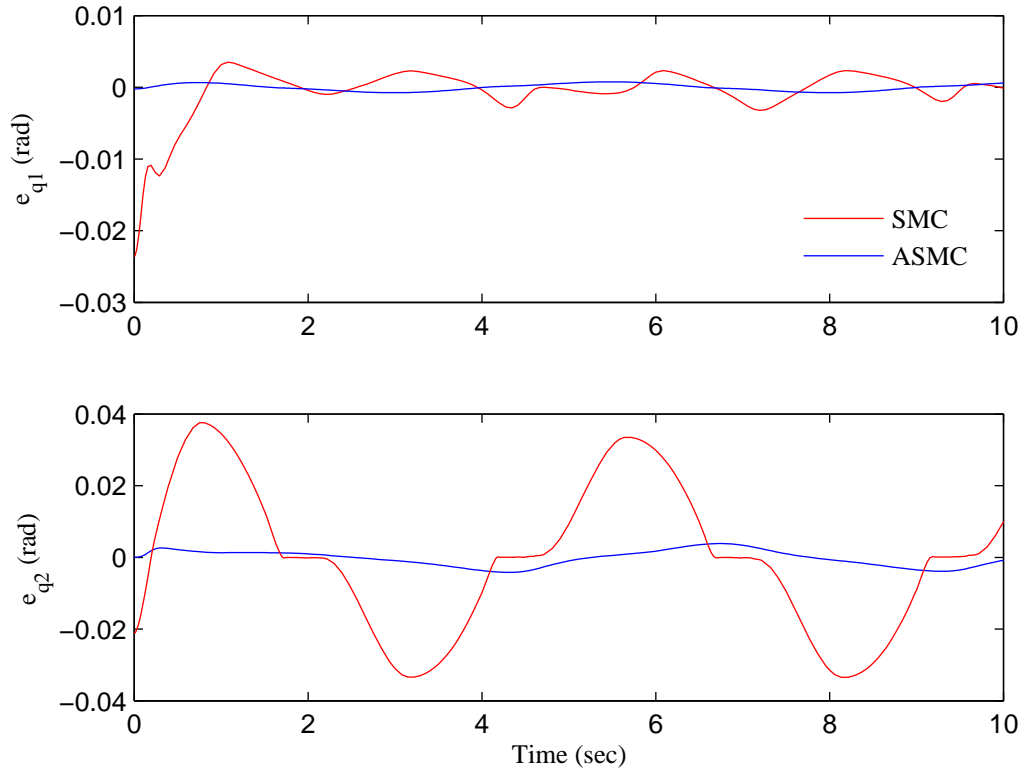


Figure 3.16.: Tracking errors obtained with SMC and ASMC for two link planar joints.

The tracking error results of the planar joint with SMC and ASMC are compared in Figure 3.16. The error percentages for both controllers are summarised in the Table 3.2.

Table 3.2.: Planar joint errors analysis with SMC and ASMC controllers.

Controller type	$e_{q1}(\text{rad})$ (%)	$e_{q2}(\text{m})$ (%)
SMC	0.1 – 0.5	2 – 5
ASMC	0.01 – 0.05	0.05 – 0.5

From the Table 3.2, joint tracking errors are obtained large with SMC based controller. The maximum errors of 5% are calculated from the joint q_2 with SMC. The ASMC controller shows better performance compared to the SMC. The max error obtained from q_1 is 0.05%. Only 0.5% error is calculated from q_2 . The joint q_2 have large errors with both controllers for sinusoidal input trajectory. The errors are large due to the damping presence at q_2 though maximum errors obtained from e_{q2} with SMC (5%) and ASMC (0.5%) are large. The overall joint tracking performances of both joints are improved with the adaptation added to the SMC controller.

3.5.6. Barrett hand simulation with SMC and ASMC

The SMC and ASMC based controllers are applied to the Barrett Hand considering the desired spread angle, $\alpha = 25^\circ$ and joint angles, $q_2 = 25^\circ$ and $q_3 = 35^\circ$. The control parameters are selected as $\lambda_f = 17$, $K_f = 220$ for SMC and ASMC. For ASMC, only the adaptation parameter is taken as $\Gamma_f = 3.3$. The joint motion tracking of SMC and ASMC based controllers are shown in Figures 3.17 and 3.18.

In Figure 3.17, the joint motions of all fingers are obtained with the SMC controller. The joint q_1 of all fingers have large overshoots (5-15%) than joint q_2 . The joint motions in Figure 3.18 are obtained with ASMC where the overshoots are less (1-8%) compared to the tracking error found with SMC. The tracking errors obtained with SMC and ASMC controllers are compared in Figure 3.19. The tracking errors obtained with ASMC exhibit with less overshoot and smooth response before settled down within 2 seconds. However, SMC and ASMC are able to track the robot joint position with unknown dynamics and payload of the robot fingers.

In ASMC, adaptation is used to estimate the dynamics. The ASMC estimation parameters of all fingers are presented in Figures 3.20 and 3.21. It is shown in these figures that the estimation process is fast to achieve the adaptation parameters of the system which are used in the parameter free regressor dynamics.

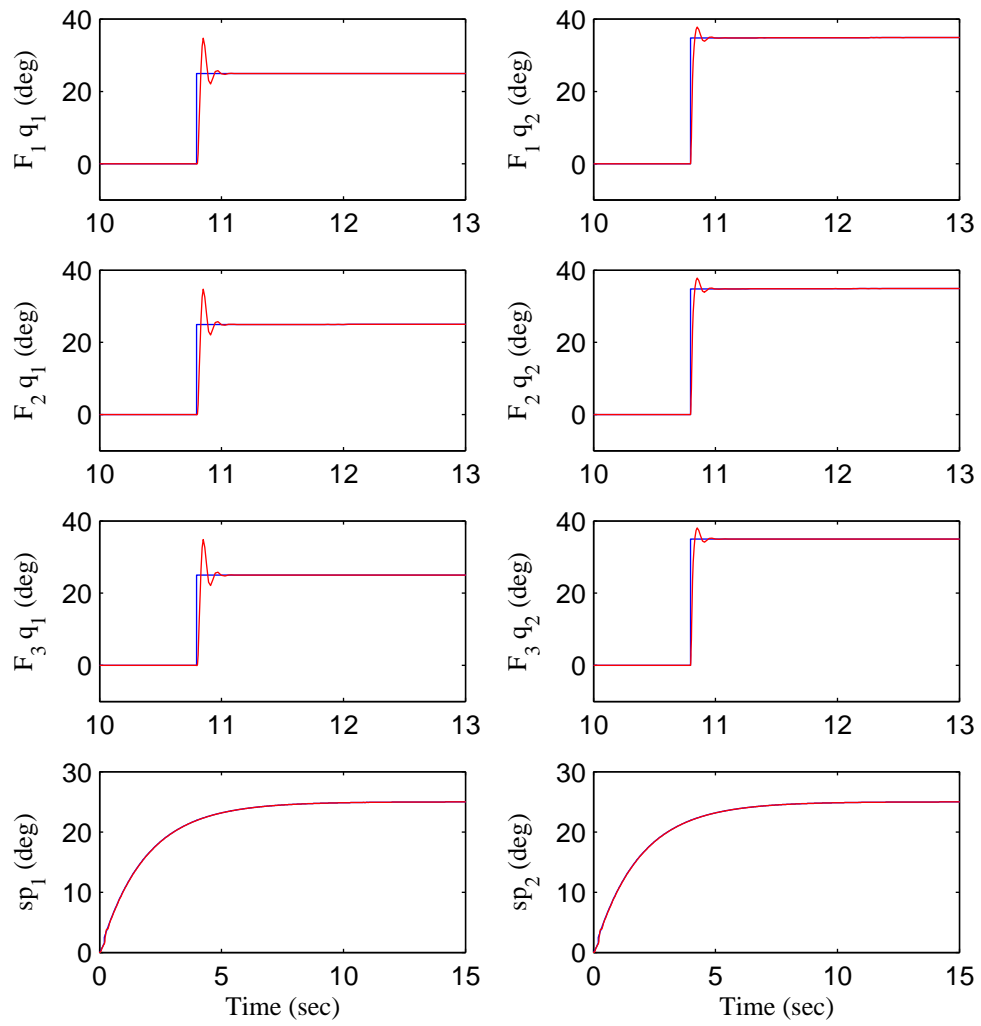


Figure 3.17.: Joint tracking results of the Barrett Hand with SMC.

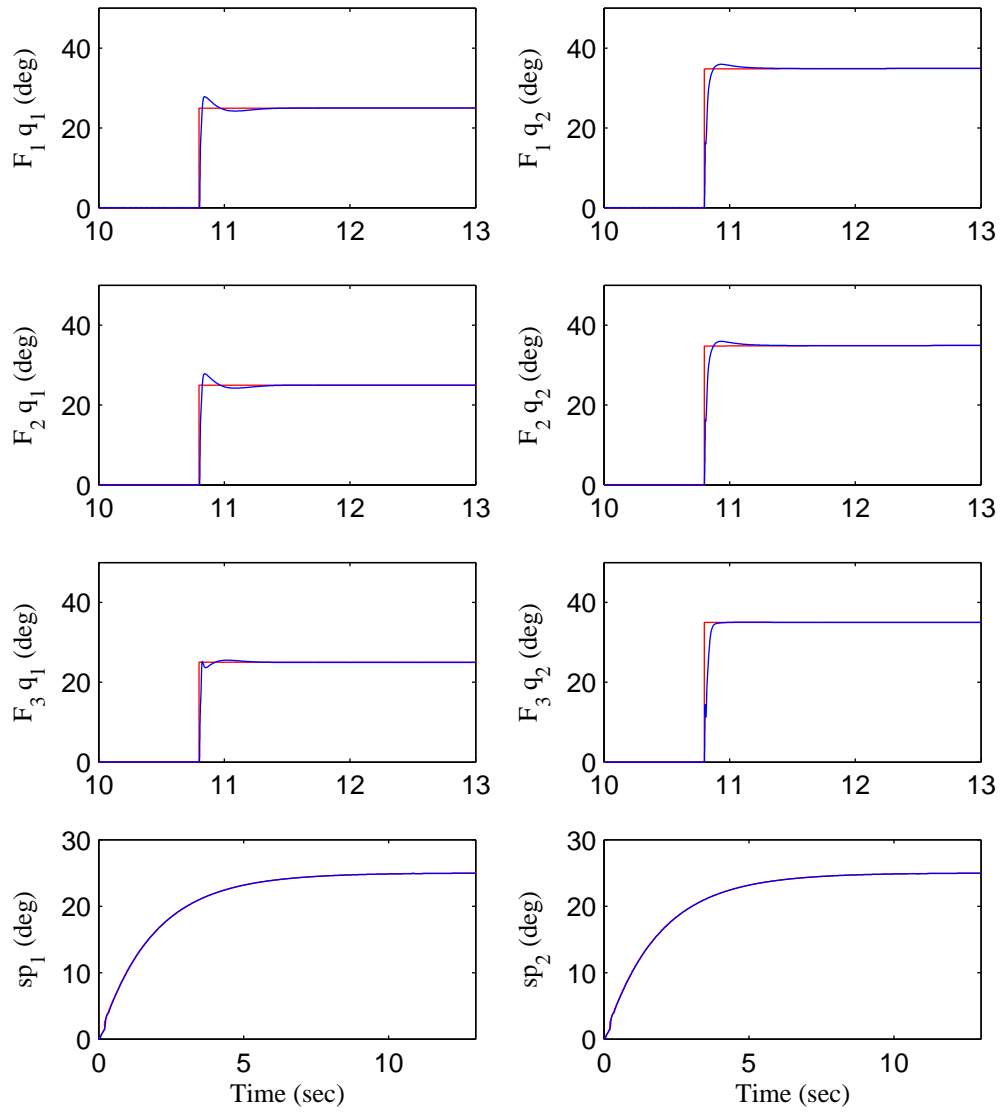


Figure 3.18.: Joint tracking results of the Barrett Hand with ASMC.

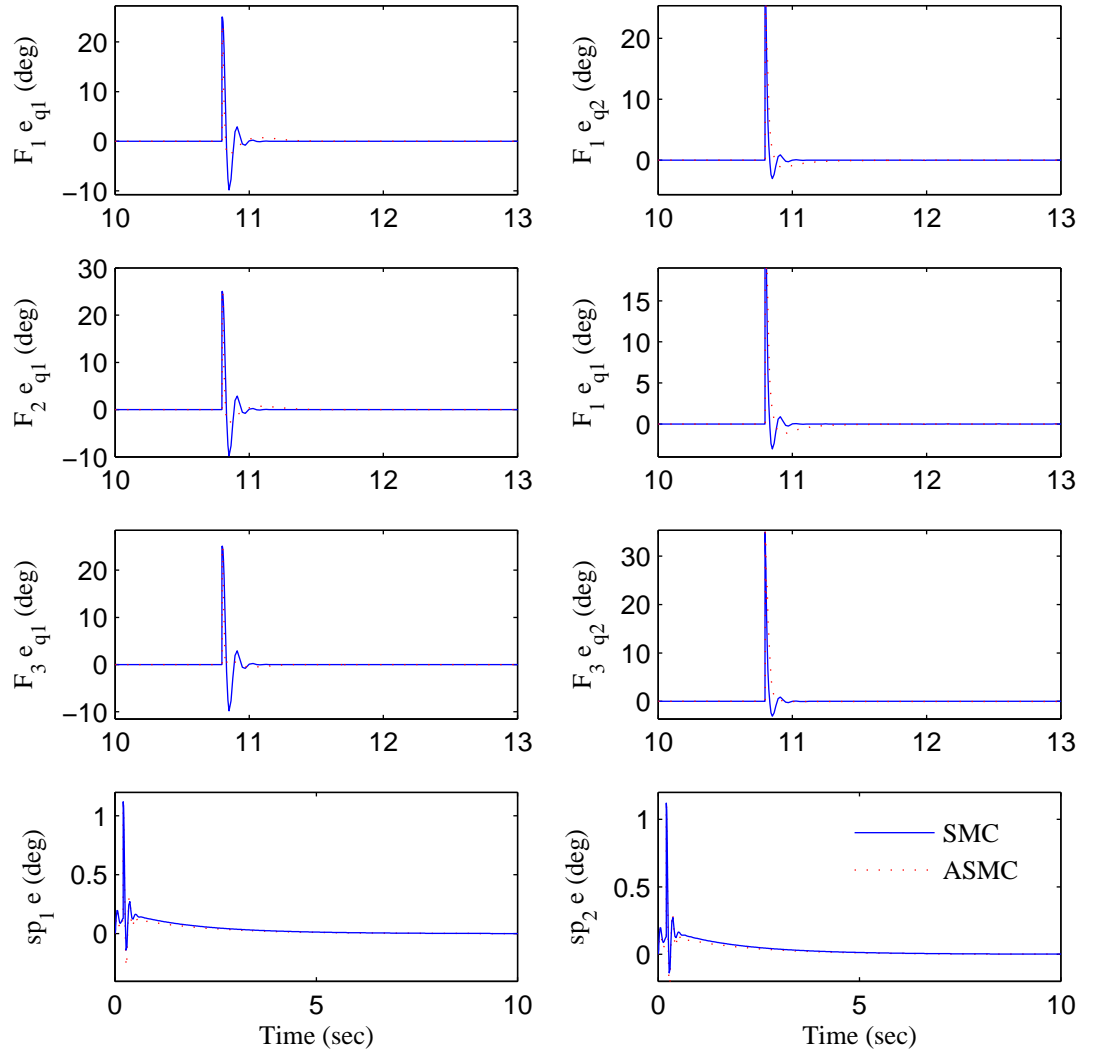


Figure 3.19.: Comparison of Barrett finger tracking errors obtained with SMC and ASMC controllers.

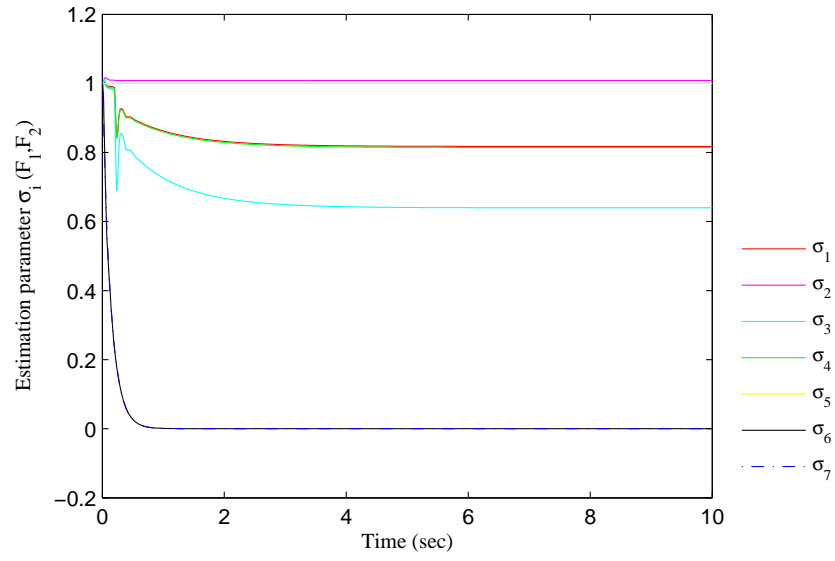


Figure 3.20.: Parameter estimation of σ_i with adaptive update law of Barrett finger F_1 and F_2 .

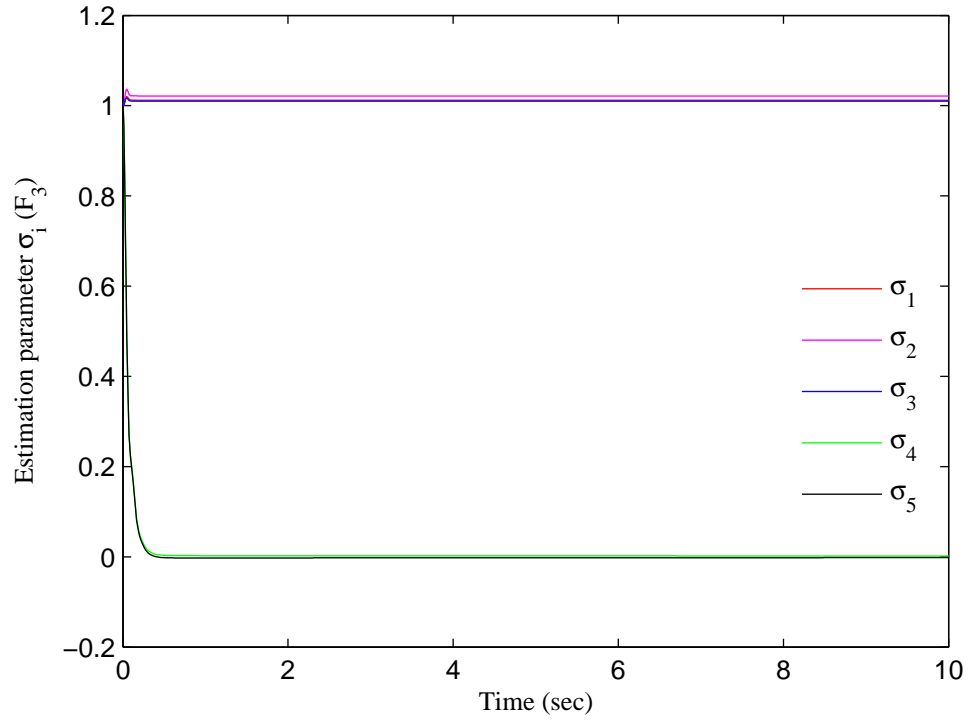


Figure 3.21.: Parameter estimation of σ_i with adaptive update law of Barrett finger F_3 .

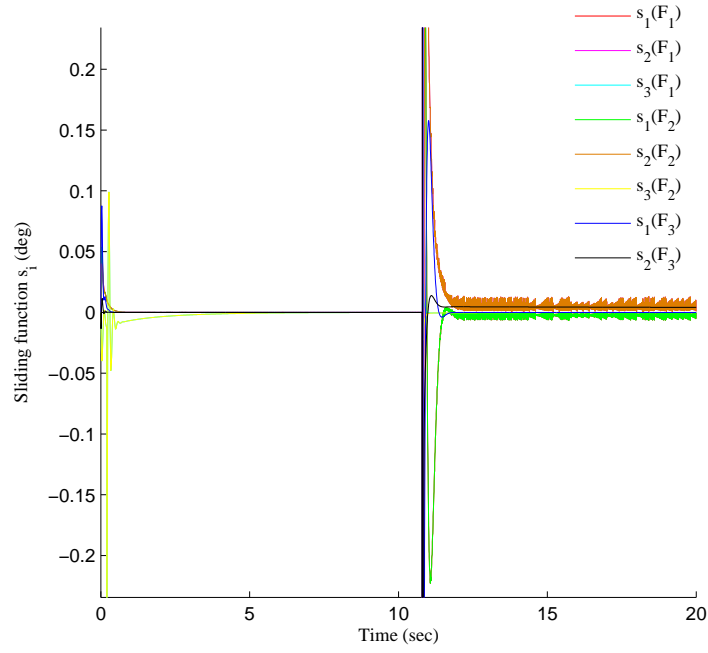


Figure 3.22.: Sliding function s_i of the Barrett hand.

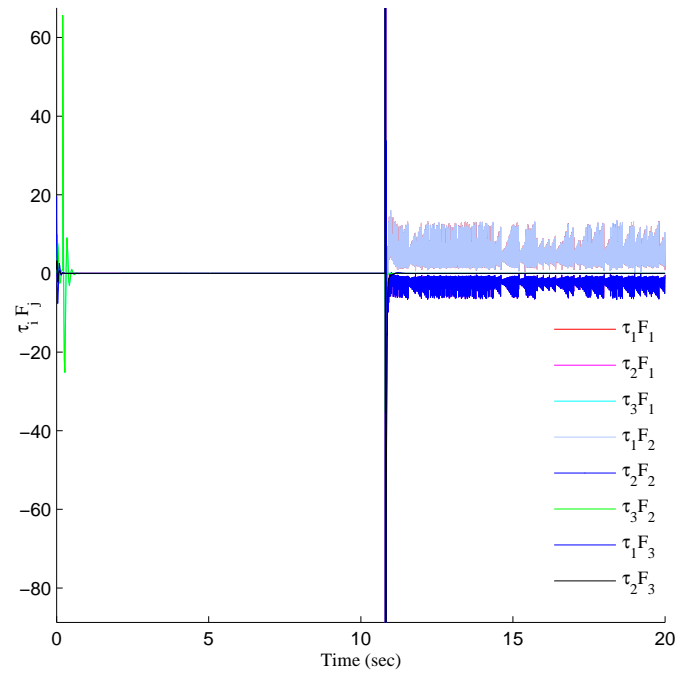


Figure 3.23.: Joint torque τ_i calculated with Adaptive-SMC law for the Barrett hand.

3.6. Higher order sliding mode

3.6.1. Background

Sliding mode control (SMC) is a successful robust control method which is capable of dealing with uncertainties and disturbances, and results in better transient performance and quicker response. In this method, the states of the system are forced to move along a chosen manifold in state space, called the sliding surfaces which are discussed explicitly in chapter 3. When the system state reaches the surface, the system becomes independent of external disturbances and parametric uncertainties. This is called the first order or classical SMC control. The SMC is proven against uncertainties but has disadvantages due to the discontinuous nature of the control law. The discontinuous nature excites unexpected high-frequency dynamics and due to the inability to estimate the boundary of the uncertainties. The chattering is the consequence of the discontinuous control signal which is the high frequency finite amplitude. Chattering problem may result in saturation and heat up the mechanical parts of the hand fingers or joint actuators. The problems of chattering and unmodelled perturbation in nonlinear system has been previously investigated [91]. Various methods have been mentioned to prevent or reduce chattering in literature. One approach is the elimination of the discontinuous nature by saturation with a compact neighbourhood of the switching surface [92],[93]. It helps to increase the tracking performance with chattering reduction. Other techniques included the design of the observer, fuzzy method for nonlinear approximation of the boundary layer of the sliding surface to reduce the chattering problem [94],[95].

Recently, higher order sliding mode control (HOSM) methods are being used to reduce the chattering effect in nonlinear system and to ensure the convergence of system state to the desired state in finite time [96]. HOSM is an extension of the sliding variable required to accelerate the system state towards the sliding surface. The idea of the HOSM is not only to limit the sliding surface but also converges higher order derivatives to zero. For an $r - th$ order sliding mode control, the $(r - 1)$ derivatives of the control input are continuous everywhere except on the sliding surface. In this case, the SMC based sliding function converge towards the higher order time derivatives of the sliding variable rather than the first order time derivative. The HOSM type controller maintains the distinctive robust structures of the sliding mode techniques and efficient in reducing chattering of the control signal.

3.6.2. Higher Order Sliding Mode structure

Levant [96] presented a method to design arbitrary order sliding mode controllers with finite time convergence for Single Input Single Output (SISO) systems in. A two part integral sliding mode control is proposed by Laghrouche et al [97] to deal with the finite time stabilization problem and uncertainty elimination problem separately. Dinuzzo et al [98] proposed another

method, where the problem of HOSM is treated as robust problem. Defoort et al [99] have developed a robust MIMO HOSM controller, using a constructive algorithm with geometric homogeneity based finite time stabilisation of an integrator chain. Adaptive control is proposed and implemented with HOSM controller in this section. For explaining HOSM, consider the nonlinear system as,

$$\begin{aligned}\dot{x} &= f(x, t) + g(x, t)u \\ y &= s(x, t)\end{aligned}\quad (3.44)$$

It is assumed that, the relative degree r of system (3.44) with respect to the sliding variable $s(t)$ is a known constant, and the associated zero dynamics are stable. The HOSM control objective is to reach $s = 0$ in finite time and hold it by discontinuous feedback function of $s(t)$ and its derivatives $\dot{s}, \ddot{s}, \dots, s^{r-1}$. The r -th order sliding mode is defined as below:

Definition 4.1:

Consider (3.44) and let the system be closed by some discontinuous feedback. Then, provided that $s, \dot{s}, \ddot{s}, \dots, s^{r-1}$ are continuous functions, and the set

$$s^r = \{x \mid s, \dot{s}, \ddot{s}, \dots, s^{r-1} = 0\} \quad (3.45)$$

is called the r -th order sliding set. The mode $s \equiv 0$ is established after finite time transient. The control appears at r -th order sliding set as,

$$s^r = h(x, t) + g(x, t)u, \quad (3.46)$$

where $h(x, t) = s^r|_{u=0}$, $g(x, t) = \frac{\partial}{\partial u}s^r \neq 0$. For some $K_m, K_M, C > 0$

$$0 < K_m \leq \frac{\partial}{\partial u}s^r \leq K_M s^r|_{u=0} \leq C, \quad (3.47)$$

which is true locally. From (3.46) and (3.47),

$$s^r \in [-C, C] + [K_m, K_M]u. \quad (3.48)$$

The closed differential inclusion is realised here in the sense of Filippov [91, 94]. The vector on the right hand side is enlarged to satisfy the convexity and semi continuity conditions. The differential inclusion does not depend on (3.44) but governed by r, C, K_m, K_M . The finite time stabilisation of (3.47) at the origin solves the control problem for system (3.44) considering (3.35). Let $i = 0, \dots, r-1$. Define,

$$\sigma_{o,r} = s \quad N_{o,r} = |s| \Psi_{o,r} = \frac{\sigma_{o,r}}{N_{o,r}} = \text{sign}(s)$$

$$\sigma_{i,r} = s^i + \beta_i N_{i-1,r}^{r-i/(r-i+1)} \Psi_{i-1,r} \quad N_{i,r} = |s^i| + \beta_i N_{i-1,r}^{r-i/(r-i+1)}$$

$$\Psi_{i-1,r} = \frac{\sigma_{i,r}}{N_{i,r}},$$

where $\beta^1, \dots, \beta^{r-1}$ are positive numbers.

Proposition 4.1. $N_{i,r}$ is positive definite i.e. $N_{i,r} = 0$ iff $s, \dot{s}, \ddot{s}, \dots, s^{r-1} = 0$. The inequality $|\Psi_{i,r}| \leq 1$ holds everywhere when $N_{i,r} > 0$. The function $\Psi_{i,r}(s, \dot{s}, \dots, s^{r-1})$ is continuous everywhere except the manifold $s, \dot{s}, \dots, s^{r-1} = 0$.

If $\beta_1, \dots, \beta_{r-1} > 0$ and chosen large in the sliding set, the controller can be defined as,

$$u_h = -\alpha \Psi_{i-1,r}(s, \dot{s}, \ddot{s}, \dots, s^{r-1}) \quad (3.49)$$

which provides the finite time stability of (3.36) and (3.37). The finite time r -sliding set $s \equiv 0$ is established in (3.48) and (3.49). Each choice of $\beta_1, \dots, \beta_{r-1} > 0$ implements a controller which is applicable to (3.49) of relative degree r . The parameter α is chosen for any C, K_m, K_M and it is negative with $\frac{\partial}{\partial u} s^r < 0$.

Considering the above properties, a second order sliding controller is expressed as follows:

$$u_A = -\alpha \left(\dot{s} + |\dot{s}|^{\frac{1}{2}} \text{sign}(\dot{s}) / (|\dot{s}| + |\dot{s}|^{\frac{1}{2}}) \right). \quad (3.50)$$

The control law in (3.50) is a continuous function of time everywhere in system (3.44) except the sliding manifold (3.45). Another second order sliding mode controller B is similar to (3.50) and have the form of,

$$u_B = -\alpha \text{sign} \left(\dot{s} + |\dot{s}|^{\frac{1}{2}} \text{sign}(\dot{s}) / \left(|\dot{s}| + |\dot{s}|^{\frac{1}{2}} \right) \right). \quad (3.51)$$

In (3.50) and (3.51), the sliding variable s is defined as the actual error of the system state. The term α is the constant sliding gain.

Consider another way to implement the controller in second order sliding mode. Recall (3.45) i.e. the control task is to keep $s \equiv 0$. For second order sliding, \dot{s} is the new sliding variable where the control is applied. In this case, the time derivative of the control \dot{u} is considered as the actual control variable which steers the sliding variable $s(t)$ to zero, ensuring $s = 0$. The advantage is that the plant control $u(t)$ becomes continuous in this mode and the chattering can be avoided. There are few robust algorithms available to solve the problem. Sub-optimal, twisting and super-twisting algorithms have been used previously in the literature [100],[101]. Recently, the super-twisting algorithm has become popular for the control of uncertain non-linear systems [102],[101]. It is also a second order sliding mode control which ensure the finite time convergence of the sliding variable and its derivative to zero. The super-twisting algorithm showed robustness in control to robot manipulators. This algorithm has the advan-

3.6 Higher order sliding mode

tage of reduced chattering compared to the sliding mode control. Moreover, it offers an simple algorithm structure which does not require the derivatives of the sliding surface function compared to other second order sliding mode controllers. It requires the measurement of the sliding surface $s(t)$ only. To implement the super-twisting algorithm, specify the second order sliding variable as,

$$s_2 = \{x \in O : s(x, t) = \dot{s}(x, t) = 0\}.$$

Define the second order sliding problem through an auxiliary system as below:

$$\begin{aligned} \dot{y}_1 &= y_2 \\ \dot{y}_2 &= \chi(x, t) + \xi(x, t)v \end{aligned} \quad (3.52)$$

where $[y_1 \ y_2]^T = [s \ \dot{s}]^T$. Equation (3.52) is defined as an auxiliary system to reduce the dynamics system to the stabilisation in finite time. The term v is the derivative of the control u where $v = \dot{u}$.

$$\begin{aligned} \chi(x, t) &= \frac{\partial}{\partial t} \dot{s}(x, t, u) + \frac{\partial}{\partial x} \dot{s}(x, t, u) \{f(x) + g(x)u\} \\ \xi(x, t) &= \frac{\partial}{\partial u} \dot{s}(x, t, u) \end{aligned} \quad (3.53)$$

Equation (3.53) is valid on $0 < K_m \leq \left| \frac{\partial \dot{s}}{\partial u} \right| \leq K_M$ and the equivalent control exists in the sliding mode $\frac{\partial}{\partial u} \dot{s} \neq 0$ and $\left| \frac{\partial}{\partial t} \dot{s}(x, t, u) + \frac{\partial}{\partial x} \dot{s}(x, t, u) \{f(x) + g(x)u\} \right| \leq \Psi$. The controller based on the super twisting algorithm is composed as

$$u_C = \int \dot{u}_1(t) dt + u_2(t), \quad (3.54)$$

where the first term is the time derivative of the control input u_C and the second term is the continuous function of the available sliding variable. Both terms are expressed as,

$$\dot{u}_1(t) = \begin{cases} -u & \text{if } |u| > u_M \\ -W \text{sign}(y_1) & \text{if } |u| \leq u_M \end{cases} \quad (3.55)$$

$$u_2(t) = \begin{cases} -\lambda_s \varepsilon_o^p \text{sign}(y_1) & \text{if } |y_1| > \varepsilon_o \\ -\lambda_s |y_1|^p \text{sign}(y_1) & \text{if } |y_1| \leq \varepsilon_o \end{cases}. \quad (3.56)$$

Sufficient conditions for convergence with the super-twisted algorithm are:

$$W > \frac{\Psi}{K_m}, \lambda_s^2 \geq 4\Psi \frac{K_M}{K_m^2} \left(\frac{W + \Psi}{W - \Psi} \right), 0 < \rho < 0.2 \quad (3.57)$$

The super-twisting algorithm does not require any knowledge on the time derivative of the sliding variable. Due to its extreme robustness, this control method is successfully applied to

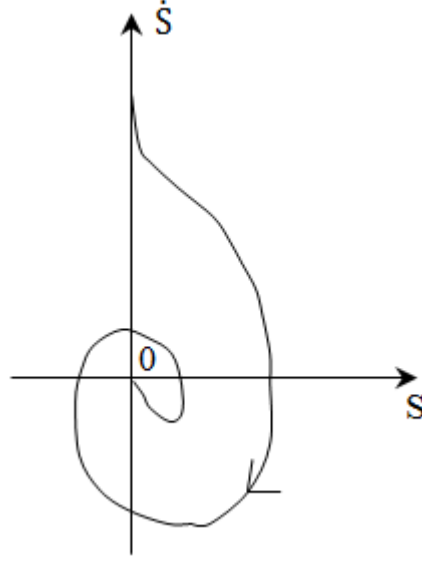


Figure 3.24.: Super twisting algorithm phase trajectory

the tracking problem in nonlinear system such as robot manipulator. The phase trajectory of the super-twisting algorithm is given in Figure 3.24.

3.6.3. Higher order sliding mode control laws

The first second order sliding controller A is chosen for the Barrett hand and is recalled from (3.50) as,

$$u_A = -\alpha \left(\dot{s} + |\dot{s}|^{\frac{1}{2}} \text{sign}(\dot{s}) / (|\dot{s}| + |\dot{s}|^{\frac{1}{2}}) \right), \quad (3.58)$$

and the second controller B is used in the Barrett hand as,

$$u_B = -\alpha \text{sign} \left(\dot{s} + |\dot{s}|^{\frac{1}{2}} \text{sign}(\dot{s}) / (|\dot{s}| + |\dot{s}|^{\frac{1}{2}}) \right). \quad (3.59)$$

For illustrating (3.58) and (3.59), consider the hand dynamics properties from (2.15) below ,

$$M_f(q)\ddot{q} + C_f(q, \dot{q})\dot{q} + G_f(q) = u.$$

Now define,

$$\dot{q} = \omega \quad \dot{\omega} = -M_f^{-1}(q)C_f(q, \omega) + M_f^{-1}(q)(u - G_f(q)).$$

If the system dynamics are assumed uncertain then, $-M_f^{-1}(q)C_f(q, \omega) \in [-F, F]$ and $M_f^{-1}(q)(u - G_f(q)) \in [G_1, G_2]$. Considering system (3.44), controllers (3.58) and (3.59) are defined when

3.6 Higher order sliding mode

α satisfies,

$$\alpha > \max \left([G_1, G_2]^{-1} \right) (\eta + [-F, F] \text{sign}(y(t)) + a),$$

where $a = 0.5\dot{s}|s|^{-\frac{1}{2}}\text{sign}(s)\text{sign}(y(y(t)))$, $\eta > 0$, $y(t) = \dot{s} + |s|^{\frac{1}{2}}\text{sign}(s)$, then the sliding variable $s(t)$ reaches zero in finite time.

The super twisting based controller C is designed for the Barrett Hand as,

$$u_C = \int \dot{u}_1(t)dt + u_2(t). \quad (3.60)$$

The first term u_1 is expressed as,

$$\dot{u}_1(t) = \begin{cases} -u & \text{if } |u| > u_M \\ -W\text{sign}(s) & \text{if } |u| \leq u_M \end{cases} \quad (3.61)$$

and the continuous term of the sliding variable u_2 is defined as,

$$u_2(t) = \begin{cases} -\lambda_s \epsilon_o^p \text{sign}(s) & \text{if } s > \epsilon_o \\ -\lambda_s |y_1|^p \text{sign}(s) & \text{if } s \leq \epsilon_o \end{cases} \quad (3.62)$$

satisfying the conditions of (3.57).

3.6.4. Higher Order Sliding Mode Control with adaptation (HOSMA)

The idea of adaptive higher order sliding arises to estimate unknown dynamics parameters online along with eliminating the problems of classical or first order SMC. The application of SMC offers robustness but does not guarantee convergence in finite time i.e. the control is applied with infinite switching action and show discontinuity in the system. The high gain control switching of SMC control input converges towards a very high value during transient time. This discontinuous fast switching action also results in chattering effect due to time delay. The chattering may introduce unmodelled higher order dynamics which leads to instability of the system. HOSM is used to remove the chattering problem by adding continuous control inside a fixed or variable boundary layer near sliding variable or considering the time derivative of control input as new control variable. The addition of adaptive control provides global stability in HOSM based system and introduced as HOSMA control. The HOSMA based controller is composed of two parts given as,

$$u = u_a + u_{hs}, \quad (3.63)$$

where $hs = A, B$ or C . The control input u_a is the ASMC law of (3.42). Only hand dynamic parameters are estimated by the adaptation which is strong in compensating unknown dynamics. Three HOSM laws for u_{hs} are derived in (3.58)-(3.62). The robustness of HOSM and HOSMA

controllers are analysed by implementing the laws in the Barrett hand.

3.6.5. HOSM and HOSMA control implementation to planar finger

The control structure of HOSM is similar to the SMC controller. The SMC introduces the chattering in the control input. This chattering may cause damage to the joint actuators. The HOSM and HOSMA controllers are applied to a planar link to eliminate chattering and improve tracking performance. The control structure of HOSM is presented in Figure 3.25.

The main feature of HOSMA is its tracking efficiency, ability to produce chattering free torque and the convergence of the system state trajectories towards equilibrium considering unknown dynamic parameters and external disturbances. For HOSMA control, the 'Linearization block' is replaced with the regressor matrix of . The control law A of (3.58) is applied for the simulation. The HOSM control parameter is selected as $\alpha = 1.3$ by trial and error. The other parameters are: reference gain, $\lambda = 0.3$, and the adaptation gain, $\Gamma = 0.7$ for HOSMA. All these parameters are obtained by trial and error.

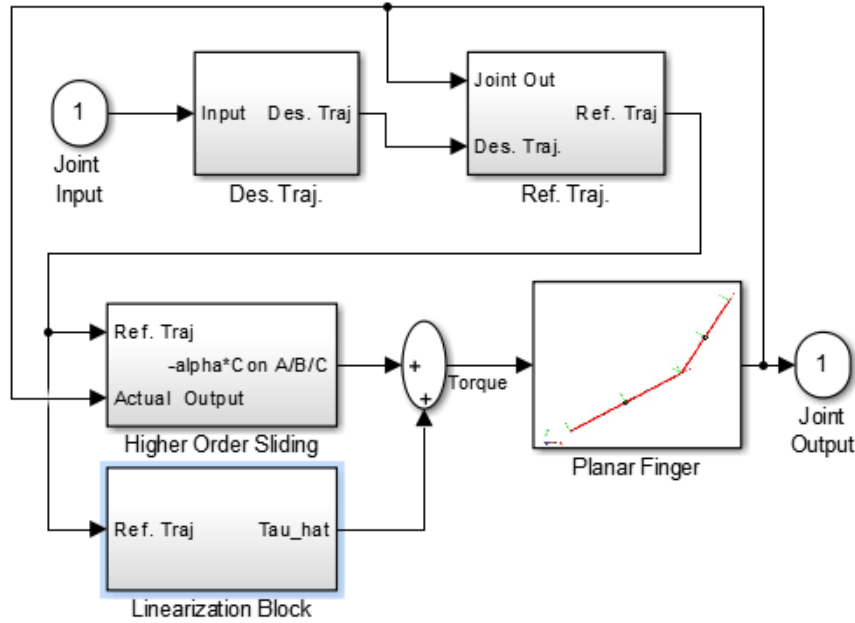


Figure 3.25.: Control structure of HOSM based control for a planar finger.

In Figures 3.26, the tracking results obtained with HOSM and HOSMA controllers are compared with SMC. The sinusoidal inputs are applied in this case. The maximum errors obtained with HOSMA are $\sim 1\%$. The error percentages are improved with HOSM ($\sim 1.1\%$) and HOSMA ($\sim 1\%$) compared to SMC ($\sim 4\%$) based controller. Random noise inputs are added to the planar dynamics and the HOSMA controller is applied to obtain the tracking responses presented in Figure 3.27. In this case, the maximum errors found from joint q_1 are

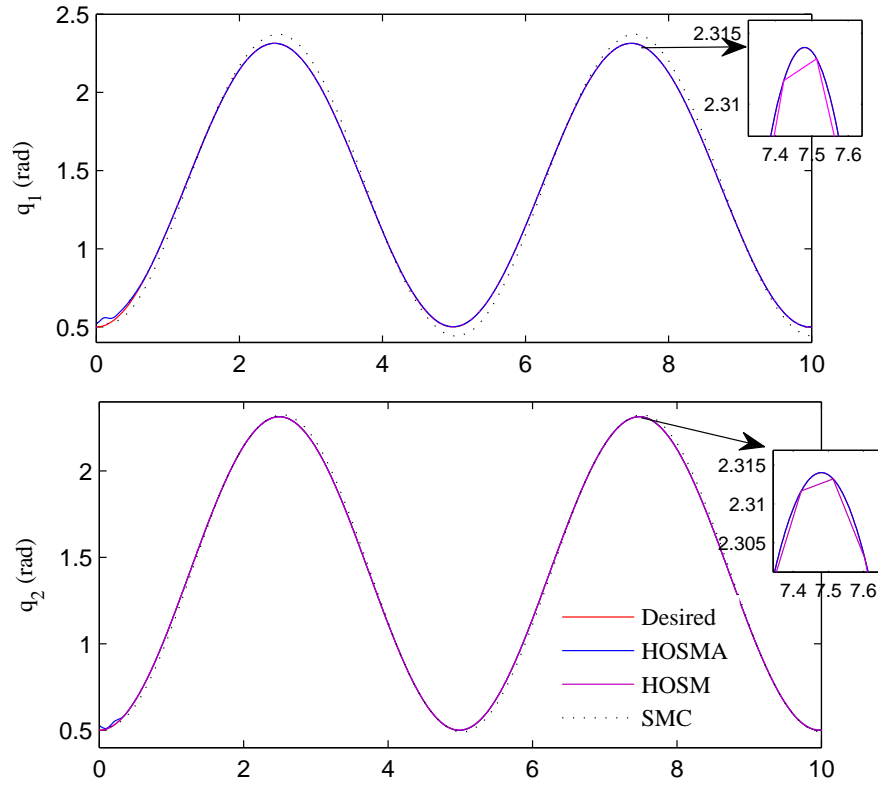


Figure 3.26.: Comparison of Joint position between HOSMA and other controllers

3% and these are particularly visible at the peak of the sinusoidal input. The error percentages obtained from q_2 are $\sim 2\%$. The overall error percentages are obtained within 1.1 – 4% for both noise and noise free dynamics.

The control torque (τ_i) and the sliding function (s_i) responses obtained with HOSM and HOSMA are compared to SMC in Figure 3.28. The chattering in torques τ_i are large with SMC due to the presence of the first order sliding mode with discontinuity. The chattering is present almost everywhere in the control torque signal. The chatterings are improved with HOSM controller and are visible in few locations of the control signals. The chatterings are not visible with the implementation of HOSMA controller and the torque signal becomes smoother. The torques are obtained within the range of -5 to $+5 Nm$. The sliding function $s(t)$ converge very close the sliding surface $s = 0$ with HOSMA. The sliding function calculated with SMC and ASMC controller do not slide along the sliding surface.

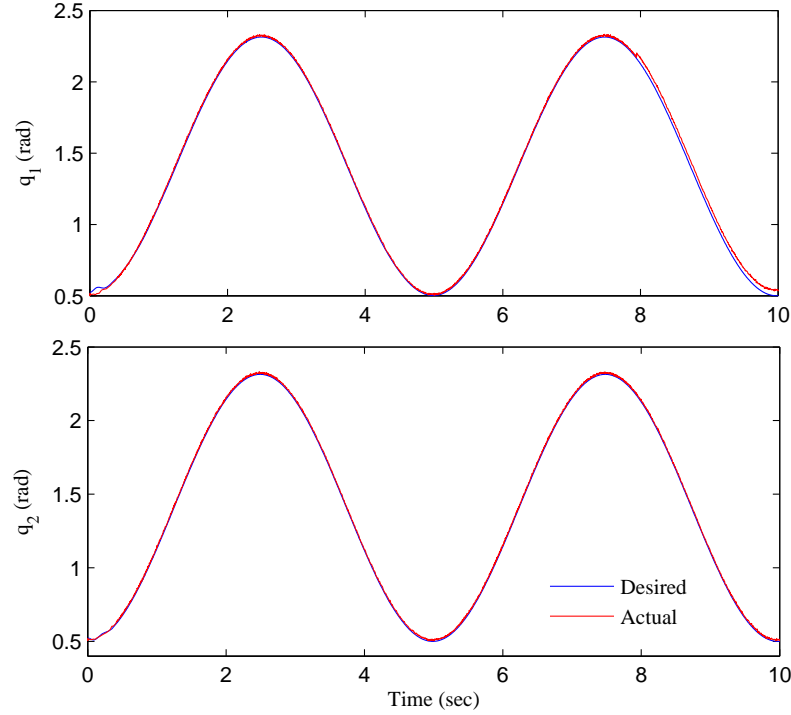


Figure 3.27.: Planar joint tracking error obtained with HOSMA controller (with noise).

3.6.6. HOSMA control implementation to Barrett finger

The controllers A , B and C derived in (3.58)-(3.63) are implemented to the Barrett hand. The Barrett hand finger F_3 is selected for the implementation of the controllers. The adaptations are added with the controllers A , B and C . The control structure is similar to the Figure 3.25. For controller A and B : the parameters are: adaptive filter gain, $\lambda = 8$, the adaptation gain, $\Gamma = 2$, HOSM gain, $\alpha = 2.1$. The super twisted algorithm based controller C parameters are: $W = 0.0015$, $u_M = 1$, $\rho = 0.1$, $\lambda_s = 2$, $\varepsilon_o = 0.001$. All control parameters for the controllers A , B and C are obtained through trial and error. The control torque u calculated of each controllers are applied to the Barrett hand SimMechanics model to see the tracking performance of the finger joints.

3.7. Results and discussions

The controllers are applied in HOSM and HOSMA mode. The Figures 3.29 and 3.30 are obtained with joint tracking of finger F_3 for desired trajectory $q_{d2} = 45^\circ$ and $q_{d3} = 15^\circ$. In Figure 3.29, the difference between actual and desired responses of q_2 and q_3 are found between $0.02 \sim 0.05 \text{ deg}$. The HOSMA based joint responses are available in Figure 3.30. In this case,

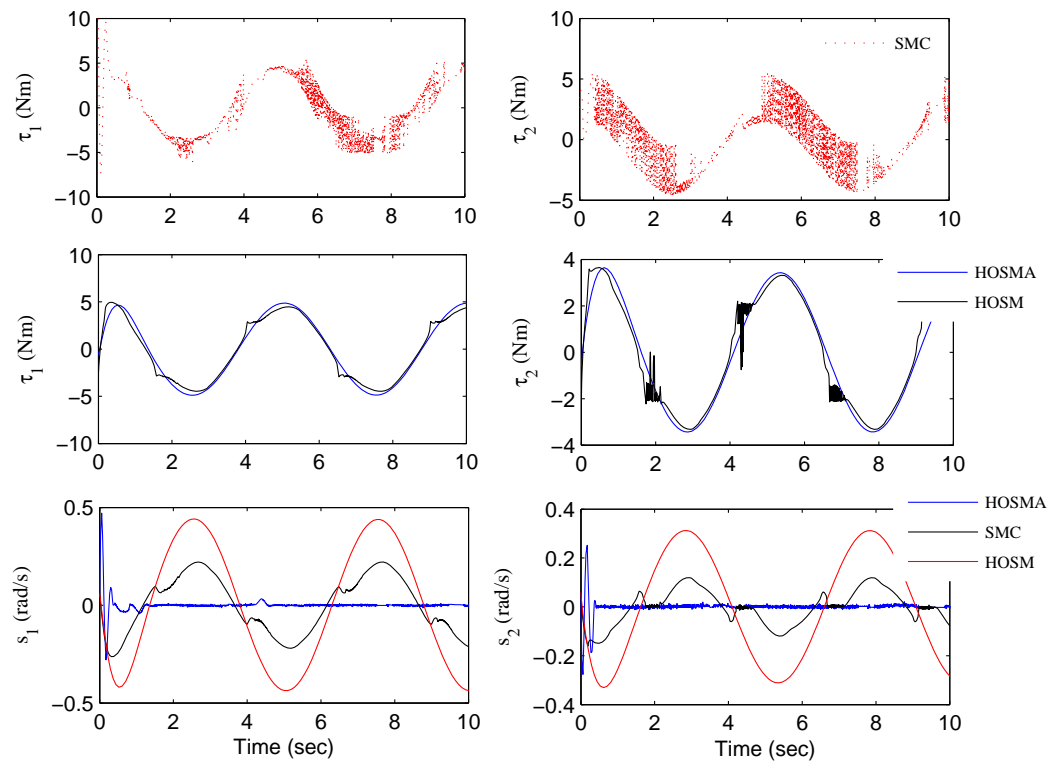


Figure 3.28.: Comparison of control signal and sliding function between HOSMA and other controllers

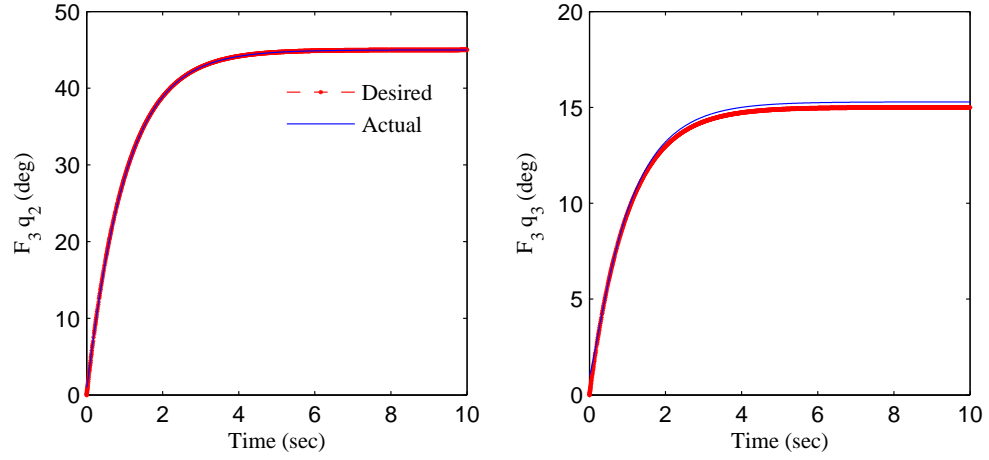


Figure 3.29.: Joint positions q_2 and q_3 of finger F_3 with controller A: HOSM

the tracking errors are reduced $0.01 \sim 0.02 \text{ deg}$ compared to the HOSM mode.

In Figures 3.31-3.33, noise free and noise based joint tracking errors are presented. The three controllers A, B and C are applied for both HOSM and HOSMA case. The responses without noises have less tracking error with HOSMA compared to HOSM mode. The tracking errors including noises are also found with less error. It implies that the noise dynamics is compensated by the controllers and the actual joint position followed the desired position. The minimum (e_{min}) and maximum (e_{max}) tracking errors for all controller are calculated and plotted in the Figure 3.34.

In Figure 3.34, the first column figures are obtained with tracking errors of q_2 and q_3 for HOSM.

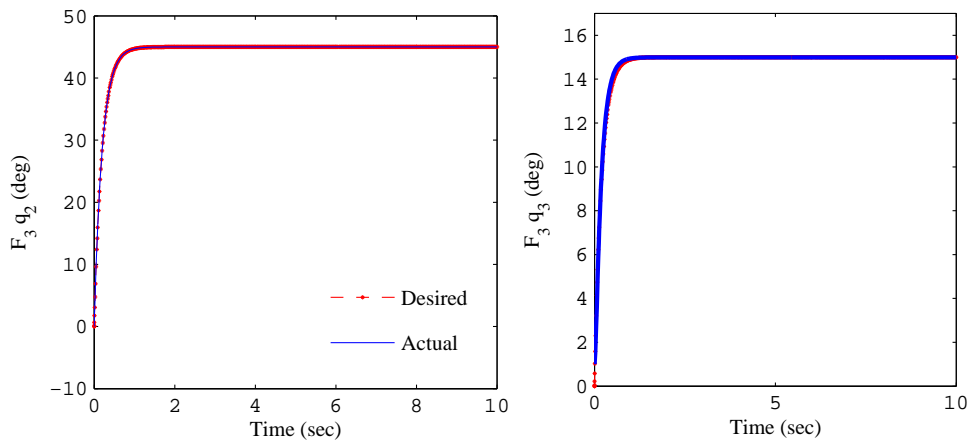


Figure 3.30.: Joint positions q_2 and q_3 of finger F_3 with controller A: HOSMA

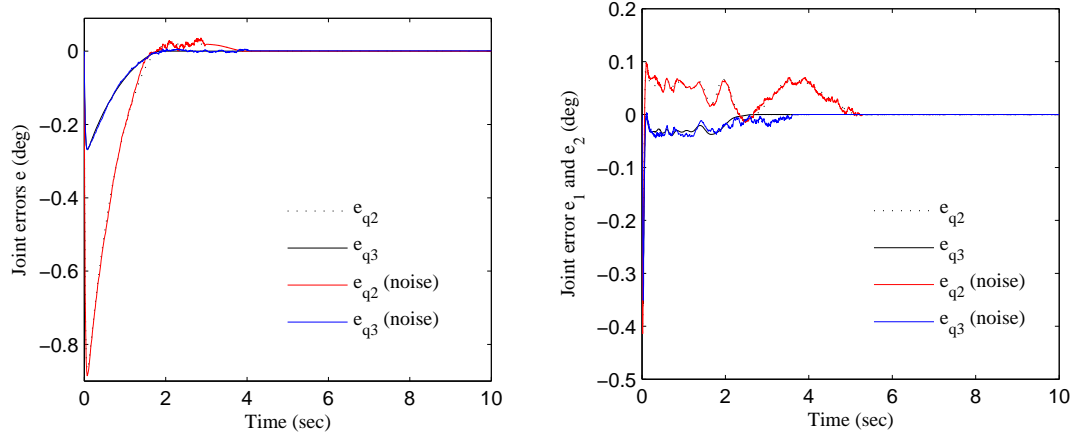


Figure 3.31.: Joint errors q_2 and q_3 of finger F_3 with controller A : HOSM (left) and HOSMA (right).

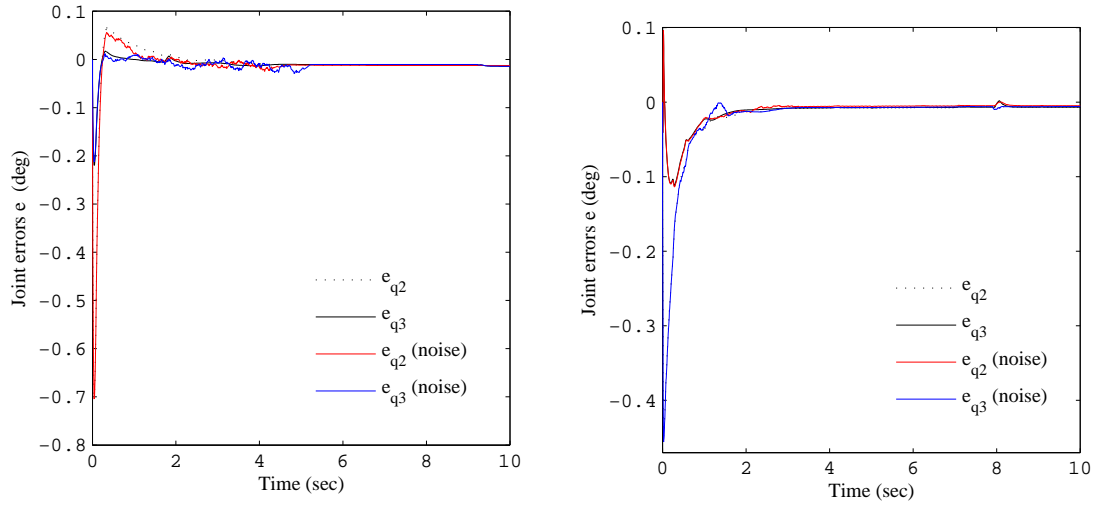


Figure 3.32.: Joint errors q_2 and q_3 of finger F_3 with controller B : HOSM (Left) and HOSMA (Right).

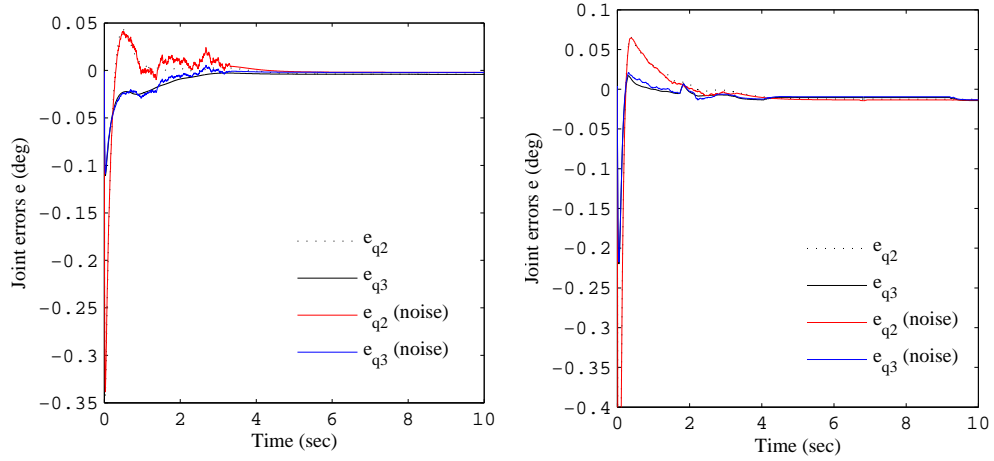


Figure 3.33.: Joint errors q_2 and q_3 of finger F_3 with super twisted controller C : HOSM (Left) and HOSMA (Right).

The second column is obtained with HOSMA based tracking errors. From the left column, the values of e_{max} are visible from noise based tracking errors. The average e_{max} is calculated as 0.7 deg . The average difference of e_{max} between actual and actual (noise) are $0.3 - 0.4\text{ deg}$. The average difference between all e_{min} are between $0.01 \sim 0.04\text{ deg}$. The differences imply that tracking errors increase with noise but errors converge very close to zero for both noise and noise free case.

The right column sub figures show the results of HOSMA with adaptation. In this case, the average e_{max} is found as 0.4 deg . The average difference of e_{max} between noise and noise free error signals are $0.22 - 0.25\text{ deg}$. The average difference between all e_{min} for HOSMA are between $0.01 \sim 0.043\text{ deg}$. The values of e_{max} and e_{min} are less in HOSM compared to HOSM for all controllers. The analysis from the Figure 3.34 depicts that HOSMA based controllers show less tracking errors for both noise and noise free finger dynamics. The HOSMA controller not only compensates the noises but also estimates the dynamic parameters of the system.

The joint torque results are presented in Figures 3.35-3.37. The chattering is improved in both modes but the control inputs are found to be smoother with adaptation. The control torques presented in Figure 3.35 are obtained through controller A. In Figure 3.35 (left), the control chattering appear until 1 second and the signal becomes smooth after that. The values of torque changes in response to the noises for both signals u_1 and u_2 but chattering reduce within 1 second. The control torques on the right are obtained with HOSMA mode. The both control signals becomes smooth within 1 second and no high chattering is visible. The noise based torques are also found with less discontinuity. The control torque responses obtained through HOSMA show less chattering compared to HOSM based signals.

The results obtained in Figures 3.29-3.36 are summarised in the Table 3.3 to see the perfor-

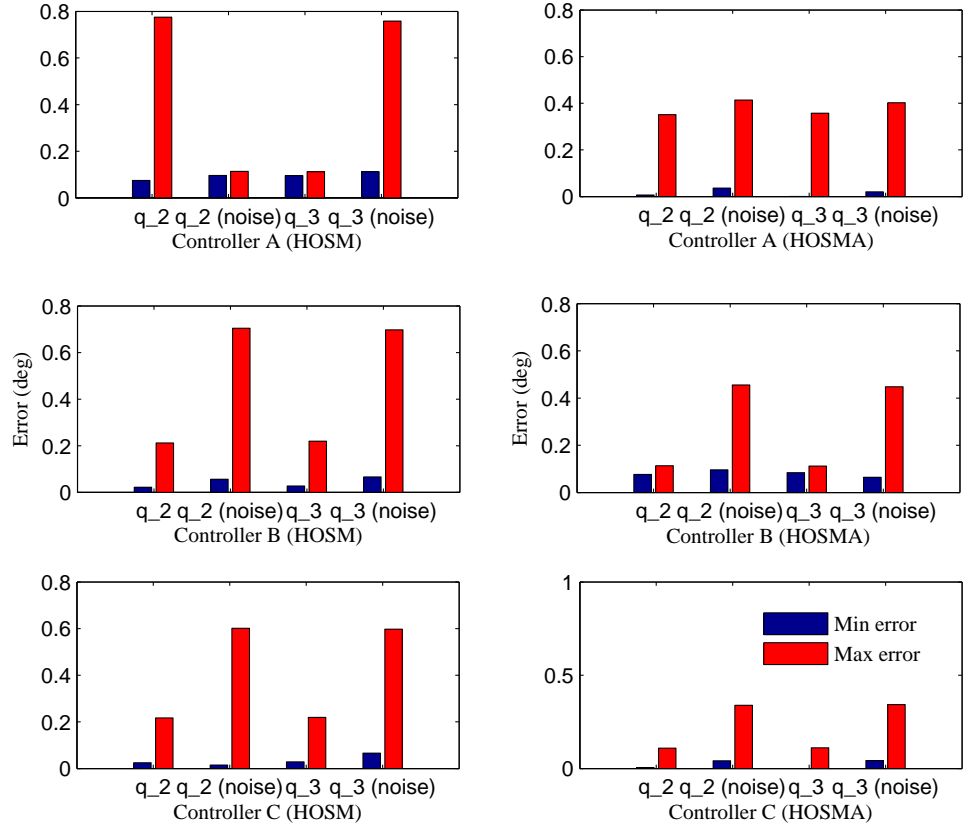


Figure 3.34.: Minimum and maximum tracking errors obtained from controllers A, B and C in HOSM and HOSMA mode.

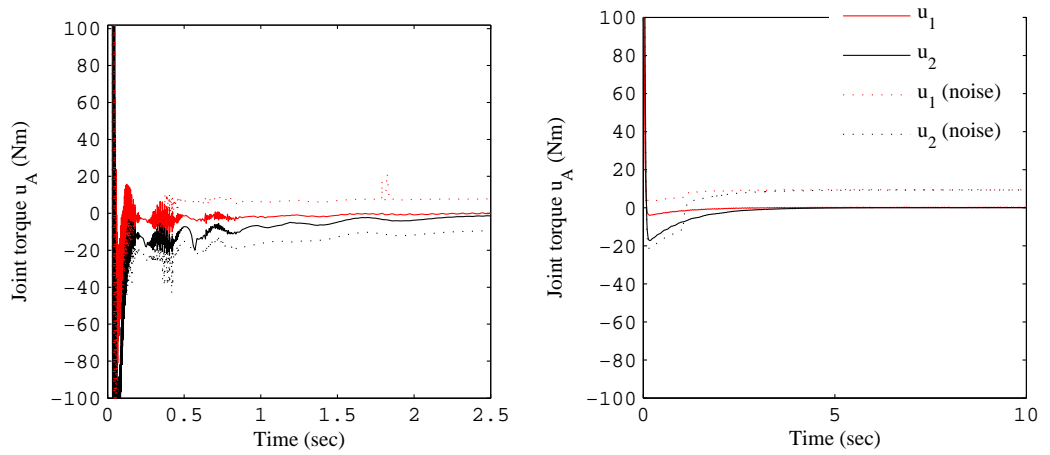


Figure 3.35.: Joint torques of finger F_3 with controller A : HOSM (Left) and HOSMA (Right).

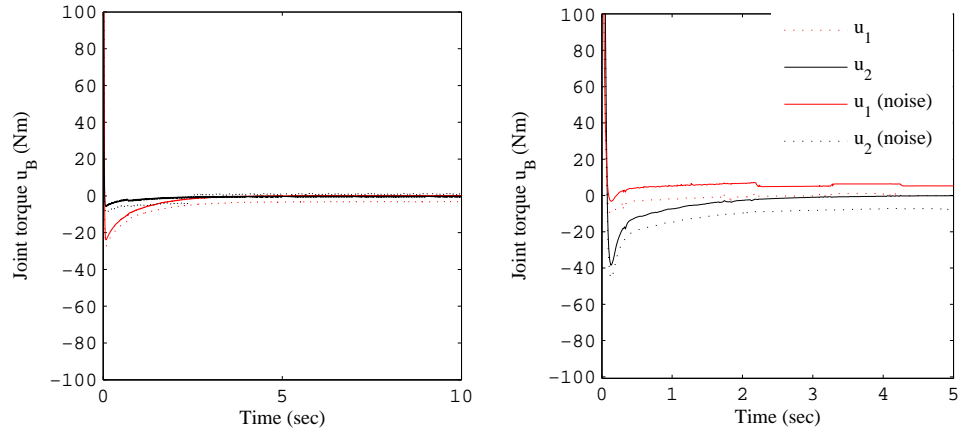


Figure 3.36.: Joint torques of finger F_3 with controller B : HOSM (Left) and HOSMA (Right).

manances of three HOSM based controllers for HOSM and HOSMA implementation.

Table 3.3.: Tracking error and torque analysis of Barrett finger F_3 with HOSM and HOSMA.

Response profiles	HOSM	HOSMA
Joint error	The error e_{max} found with noise and for the noise free case was 8% and 5% respectively. The convergences of the actual trajectories to the final values were not affected by noise. This implies that the minimum errors e_{min} are found between 1 – 2% of the desired response. In spite of the little variation with noises, the controller performances are good in achieving reference position.	The controllers are tested in HOSMA mode with adaptation. The errors obtained from noise and noise free cases are found with an average value of 4.5%. For cases with noise, errors do not increase as observed with HOSMA. The consistent performance of HOSMA based controllers are proved by the steady error percentage. The minimum error e_{min} was also found between 0.5 – 1.5% of the desired trajectory.
Control torque	The chattering can be seen at the beginning (before 2 seconds) of the simulation. The noise added to the dynamics puts more effort in control signals but torques are obtained smoother for both noise and noise free cases.	The control signals are obtained smoother in this mode. The chattering is compensated with noise and with the noise free case for the simulation. The control torque varies due to the noise. The extra control efforts cause small chattering.

According to analysis of the the tracking errors and the control torques from Table 3.3, the

proposed three controllers implemented in HOSM and HOSMA mode show robust tracking performances against uncertainties and noise. The controllers exhibit very less chattering in HOSM and HOSMA mode.

The robotic finger dynamics contain non linearities and unknown joint friction parameters which restrict the robot dynamics to achieve the desired location in real case. The SMC based controller have robust nature to control a non linear robotic finger but high frequency chattering cause problems to the joint actuators. The SMC controller is applied in higher order to reduce the chattering and deliver steady tracking performance. An adaptation is included with HOSM to estimate the imprecise dynamic parameters of the finger. The both HOSM and HOSMA based controllers are applied to the SimMechanics Barrett finger model. The noises are added to bring uncertainties similar to the real robot fingers. The results obtained from the simulations show that the implemented controllers are able to provide chattering free torque and consistent tracking performance against all structured and unstructured uncertainties.

3.7.1. Comparison of the controllers

In this chapter, three type of controllers are proposed and implemented for joint control of the robot hand. Two type of models are used in this case: the two link planar finger and the Barrett hand (Modelled with CAD design of real Barrett hand). The CTC methods are first applied to the Barrett hand to see the tracking performance. This controller is unable to deal with uncertainties exist in robot hand dynamics. Moreover, tracking problems play an important role in high precision manipulation with multi-fingered hand. The SMC and ASMC controllers is proposed and implemented to reduce the tracking errors but introduce chattering in control signal. The HOSM and HOSMA based controllers are proposed and implemented to achieve steady tracking performance and reduce the chattering of control signal considering the robot uncertainties. The controller performances are compared and summarised in the Table 3.4.

Table 3.4.: Comparison of the implemented controllers applied to the planar and the Barrett finger.

Controller name	Features and performance
CTC	The CTC method is well suited to known system dynamics. The non linear dynamics of the robotic hand is assumed to be known in this case. The CTC based controller is formulated to cancel the non linearities and linearise the Barrett hand dynamics. The proportional and the derivative gains are applied to control the joint positions and to achieve the desired joint values. The controller is applied to both simulink and SimMechanics based Barrett hand to assess the tracking performance of the joints. The tracking errors are converged to zero exponentially with quick settling time. The SimMechanics model appeared to be slower than simulink model due to the presence of more components but have similarities to the real Barrett hand. The controller is unable to estimate the hand dynamic parameters precisely considering uncertainties.
SMC and ASMC	The SMC controller steers the state trajectory to the equilibrium point of the system regardless of the uncertainties. The SMC based controllers are derived for planar and Barrett hand. The tracking error percentages are found less ($\sim 5\%$) with planar link. The overshoots are found large ($5 - 15\%$) for Barrett hand with SMC. The ASMC controller is applied to the hand to improve the tracking performance. This helped to reduce the overshoot and to decrease the tracking errors between ($2 - 8\%$) of the desired response. The control chattering found for both planar and Barrett hand are responsible of damaging the joint actuator or the object in practical situation.
HOSM and HOSMA	The HOSM based controller is proposed to integrate with SMC mode. This controller is not only able to steer the nonlinear system state trajectories along the higher order sliding surface but also reduces the control chattering. The HOSM is applied to the planar and the Barrett hand. The tracking errors obtained with the planar link are $\sim 1.1\%$ for HOSM and $\sim 1\%$ for HOSMA. The control chattering is reduced more with HOSMA mode. The Barrett finger has 8% (with noise) and 5% (without noise) tracking errors obtained with HOSM and HOSMA respectively. The control chattering of the Barrett finger is reduced compared to the SMC based controllers. The results obtained with HOSMA show efficiencies in compensating noise, estimating unknown dynamic parameters and reducing chattering with less tracking errors and sustain robustness. The performance of the HOSMA against uncertainties would be suitable to apply into the real robot hand.

4. Interactional kinematics, dynamics and force control of the Barrett hand

4.1. Introduction

The general aim in grasping and manipulation tasks is to grasp an object and manipulate it by a robotic hand. There will be no usage in an articulated multi-fingered hand if it cannot deal with an unknown or dynamically changing environment. It also needs to be able to cope with varying environmental properties such as soft objects (e.g. orange, rubber ball) to hard objects (eg. cylinder, metal box). Predefined environment properties are not suitable when the hand is used for grasping and manipulation. The visual recognition of the object information, visual tracking and three dimensional pose estimation are the most advanced technologies in terms of using external sensors to assist a robot to become familiar with the environment, environmental changes and uncertainties. Possible scenarios which can be presented range from manipulation tasks in nuclear environment, tools fixing tasks in space-station, search and rescue tasks in hazardous environment, to name but a few. Another growing area confirmed by the latest industrial activities, is the application of force sensors mounted on the hand to allow regulation of the interaction forces between the hand and the environment. Force sensors are mounted on fingertips of each finger of a robot hand and able to measure the forces exerted on the fingertips. These measurements can be used as feedback for developing controllers. Most of the current industrial hands consist of force sensors to estimate the contact force from the environment and to control accordingly. Researchers have been working on force control application for the past 30 years [103],[104],[105],[106]. The original aim was to limit the maximum amount of forces that a robot manipulator could exert on the environment. However, integration of force sensors on robotic end-effectors have been available for the last few years [107, 2]. Classical force control based strategies are not feasible to compensate the hand object interaction due to the absence of online adaptation mechanism or inability to determine the compliant behaviour of the coupled robot-object during interaction. Modern research has been focusing on control strategies like impedance or adaptation based control [108],[109],[110]. This type of method converts the robotic hand into a compliant one with feedback control.

Interactional modelling is the requirement to implement any control for manipulation. When

a random object is grasped, it is not straightforward to pre-estimate its geometric and stiffness properties. The interactional model determines the contact force between the robot and the environment. Different types of object stiffness can be applied to see the changes in the contact forces. The accuracy of the contact force depends on factors such as the correct structure of the model, the geometrical shape of the object and the robot hand fingertip, the stiffness of the object and the fingertip. The interactional model is the basis of post-grasp control of the multi-fingered robotic hand. When the contact is made at the grasp state, the fingertip position becomes constrained along certain directions by the object and a force is generated. The compliance behaviour of the robotic hand and the object is used to control this force. This force allows the object to be manipulated at a desired configuration.

In this chapter, the general interaction between the multi-fingered Barrett hand and the object is modelled to calculate the contact force. To accomplish grasping and manipulation tasks, the contact force needs to be optimised. If the force is not controlled, it may cause damage to the object and the hand and its joint motors resulting in unsuccessful grasping and manipulation activities. Therefore, control strategies are extended to the force domain with impedance techniques in this chapter. Several established impedance methods are presented to optimise the contact force. Position and force based impedance controllers are developed and implemented on the Barrett Hand. The ability to interact with an unknown object's environment is still a challenge to be successfully overcome by the force sensors. For a well known environment, force sensors can provide correct measurement. When the environment is partially known or unknown, force sensors itself is not enough to deliver accurate force feedback. The integration of adaptation mechanism allows the force sensors to be familiar with the environment at every state (from beginning to the end) of the tasks. Considering these, adaptation based impedance control is implemented to assess the performance of the Barrett hand in dealing with contact compliance.

4.2. Contact mechanics

There are different ways of modelling the interactions between two bodies. Most of the controllers focus on contacts in a static environment. Grasping and manipulation of rigid bodies can be well presented by static contact. The interaction between a Barrett finger and a fixed environment is modelled in this section. The fingertips of the hand are assumed to be rigid for modelling but the object and environment are considered with varying stiffness properties. It is assumed that the position, orientation and shape of the object are known through the sensor attached to the hand. The interaction is modelled to show how the contact force and joint torque can be calculated before and after contact considering different stiffness values of the object. After contact, the reaction forces exerted on the finger are sensed by the fingertip. These reaction forces vary according to the object stiffness.

The two link revolute Barrett finger F_3 are considered to illustrate the contact model. The dynamics of the finger is given below,

$$M_f(q)\ddot{q} + C_f(q, \dot{q})\dot{q} + G_f(q) = \tau, \quad (4.1)$$

where $q(t), \dot{q}(t), \ddot{q}(t) \in \mathbb{R}^2$ represent the angular position, velocity and acceleration of the finger respectively. The terms M_f, C_f, G_f are the matrix elements of the robot hand dynamics and τ represents the joint input torque. The Cartesian position of the fingertip is denoted by $X_f = [x_f, y_f, z_f]^T \in \mathbb{R}^3$. This can be linked to the joint space through the kinematic relationship as follows,

$$\dot{X}_f = J_f(q)\dot{q}, \quad (4.2)$$

where $J_f \in \mathbb{R}^{3 \times 2}$ is the manipulator Jacobian. The compliant robot-object system structure is given in Figure 4.1. The term z_f and z_c are the fingertip and object position respectively in the z direction with respect to the hand base frame C_b . The position vector r_f and r_o maps the coordinate to X_f and X_c .

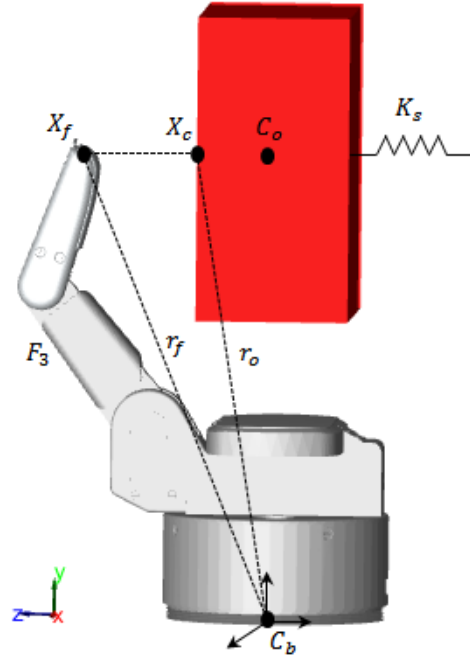


Figure 4.1.: Contact mechanics between F_3 and an object with linear.

The object is considered as a mass-spring system with the stiffness values between, $K_s = 10 - 50 \text{ kN/m}$. The unforced object dynamics is given as,

$$m_o \ddot{z}_o + K_s(z_o - z_c) = 0, \quad (4.3)$$

where $z_o(t)$, $\dot{z}_o(t)$, $\ddot{z}_o(t)$ represent the displacement, the velocity and acceleration of the object in the normal direction respectively. The object mass is m_o and z_c is the initial unperturbed position of the object. It is assumed that $z_f(t)$ and $z_c(t)$ are bounded as follows,

$$\zeta z_f \leq z_o \leq \zeta z_o, \quad (4.4)$$

where ζz_f is a constant which is determined by the minimum coordinates of the finger F_3 along z axis and ζz_o is also a positive constant. The lower bound of z_f is in accordance with the finger geometry and the upper limit of z_o is based on the actual fact that the object is attached by spring with stiffness ranged from .

Consider the fingertip of F_3 as it makes contact with the object. The impact between the finger and the object occurs only when $z_f \geq z_o$. This impact generate equal and opposite force between the coupled system. The force exerted on the object is represented as,

$$F_o = K_s \Lambda(z_f - z_o), \quad (4.5)$$

where the term $\Lambda(z_f, z_o)$ is defined as,

$$\Lambda = \begin{cases} 1 & z_f \geq z_m \\ 0 & z_f \leq z_m \end{cases}. \quad (4.6)$$

It is noted from (4.6) that, the impact force F_o depends on the values of Λ . F_o is zero before the contact is made. When, the F_3 fingertip hits the object, the first condition of (4.6) becomes true and impact force on the object is determined. Considering (4.5), rewrite the unforced object dynamics of (4.3) as,

$$m_o \ddot{z}_o + K_s(z_o - z_c) = F_o. \quad (4.7)$$

Similarly, the impact force F_c exerted on the fingertip produces torque τ_d denoted by,

$$\tau_d = J_f^T F_c, \quad (4.8)$$

where J_f is the Jacobian of (4.2). The equation (4.8) is called the Jacobian transpose relationship which converts the exerted fingertip forces to the equivalent torques by the transpose of the robot Jacobian. The dynamic model of the finger link interacting with the object is now expressed as follows,

$$M_f(q)\ddot{q} + C_f(q, \dot{q})\dot{q} + G_f(q) = \tau - \tau_c. \quad (4.9)$$

Equation (4.9) is the model of the system with interaction dynamics where τ is the control input torque. It is applied considering different forces and interactions based on the control approach

to see the affect at the time of collision between the robot finger and the object. There is another assumption which needs to be mentioned for subsequent control development of (4.9). The minimal singular value of the finger Jacobian is larger than $\delta > 0$, a positive constant such that the maximal of the inverse of Jacobian is known *a priori*. By this, kinematics singularities are avoided.

The remaining sections of this chapter provides the formulation of interactional control problem and few classical and adaptive based methods for controlling the interactional forces.

4.3. Problem formulation for force control

It appears from the previous section that the implementation of force control depends on the contact force generated from hand finger-object interaction. The contact force is a function of fingertip and object position. These positions are expressed in Cartesian coordinates. Therefore, dynamics of (4.9) is required to transform into the Cartesian workspace format. Using (2.37), (4.9) can be re-written as,

$$M_x(q)\ddot{X}_f + C_x(q, \dot{q})\dot{X}_f + G_x(q) = F - F_d, \quad (4.10)$$

where X_f is the position and orientation based vector of fingertip. The term F is the driving force of the hand and F_{dc} is the contact force vector. Now the interaction based control problem can be formulated from (4.10). The problem is to find the joint torque of the finger τ considering constrained force F_c generated from object environment such that, the position of the fingertip is maintained i.e. follows commanded trajectory during contact and the produced contact force is optimised. The problems are mathematically expressed below when the object interacts with the robot finger.

Given a desired contact location trajectory X_{df} each fingertip of the robotic hand X_f should track X_{df} . The position of the fingertip will be tracked as,

$$\lim_{t \rightarrow \infty} X_f(t) = X_{df}(t), \quad (4.11)$$

when the contact force F_c follows the desired contact force F_d . The force problem is given as,

$$\lim_{t \rightarrow \infty} F(t) = F_d(t). \quad (4.12)$$

It is mentioned that, (4.11) and (4.12) is only valid when contact is made. The two problems are related. The actual position trajectory of the fingertip is an input element to determine the contact force. The fingertip position depends on the value of the contact force. When the contact force is optimised, the fingertip position does not change abruptly and stays very close the desired one.

4.4. Force control techniques

The control problem of interactional dynamics lies in the calculation of contact forces and regulates various tasks such as grasping and manipulation, peg in hole and deburring. The contact force modification depends on the motion of the robot and object at the point of interaction. In general, if the relationship between them is known, the force can be adjusted either in the motion or force control loop. For this reason, correct interaction forces prescribe the appropriate adjustment needed to control the robot-object coupled system. Compliance is considered as the capability of the robot link to respond to this interaction forces. This represents a wide range of different control techniques in which the motion of the hand fingertip can be modified by the contact forces [106],[43],[110]. The problems experienced in controlling the compliant based robot motion have been substantially analysed and different methods have been suggested and implemented in the past. The application of these control methods depends on two types of compliant motion.

1. Passive compliance: The robot end-effector position is regulated by the contact force indirectly with this mode. This is due to the internal compliance structure of the hand, joints or other compliant parts.
2. Active compliance: In this mode, the compliance appears by implementing a feedback force to achieve a reaction of the robot. The implementation of the feedback can be done either by controlling the interaction force or generating specific compliance at the end-effector.

In active compliance, force feedback is applied to see the response of the robot. This is the strategy used in implementing the interaction between a robot link and the environment. Robots containing internal mechanism with compliance can be ignored. Robot dynamics can be mathematically modelled without the internal mechanism and fingertip motion of the robot can be easily determined. Active compliance can be categorised into two control groups:

- i) Hybrid position or force control: In this technique, the position and forces of the coupled system are controlled by two orthogonal subspace expressed in a constraint frame. Each of the subspace is restricted to a particular domain. For example, a manipulator is required for a deburring task. In free space (unconstrained), the motion subspace activates the position control mode. When a force is applied for deburring at the task frame, the control switches to force domain. This approach is first referred by Raibert and Craig [104]. Another method described by Vukobratovic and Ekalov [111], is called the unified control of the position and the force. This study presented a dynamic approach to control the force and position simultaneously.
- ii) Impedance control: This is a form of interactional control which regulates robot's dynamic behaviour. It is based on the relationship between the constrained forces and the distance

4.5 Impedance control (IMC)

between the robot fingertip and contact position on the object surface, to regulate the mechanical impedance of the compliant dynamics [108]. It is expressed as implementing a control method so that the interaction forces drive the position error of the end-effector in accordance with the desired impedance of the link. It is based on the position control. The target position command and actual measurement required to close the feedback loop. Also, contact force measurements are used to acknowledge the desired impedance characteristics.

In this section, two compliance categories are described. It is noted that, active compliance is applicable in grasping and manipulation tasks. Different control techniques are discussed based on the active compliance. The drawbacks of the hybrid position or force control are its switching problem between the position and the force control domain. Integration of switching for any controllers require infinite energy in the control signal. The motor assembled to the robot joints produce only limited torques which are unable to deal with high frequency control signal. Also, the hybrid method is suitable where consistent position or force control is required one after another. In a grasping and manipulation task, direct force control is not needed for continuously, compliance is used instead to accomplish the tasks. Therefore, impedance based control is found as a suitable option to be integrated in for grasping and manipulation tasks due to its ability to regulate the interaction force from the compliance.

4.5. Impedance control (IMC)

A scenario is presented to illustrate the idea behind impedance control (IMC). Assume an assembly robot is required to pick up an element, shift in free space and carry it whilst in contact with the environment (kinematic constraint), position it in constraint direction and release it, before finally returning to collect another. The control objective at the first step is position or force control. In the second mode, only the position control is needed. In the last stage, an active compliance is present between the robot and the environment which changes the control objective. The objective at each mode is different to the others and should comply with the tasks assigned. Hence, all tasks with regards to different objectives present a challenge. For three different controllers, it is difficult for the system to decide which controller will be in use at each task frame. Also, the transition between each control subspace is not straightforward and requires a large amount of energy. Considering the issues of different objectives for multi controllers and the transition of controllers between different control domain, a single controller can be used instead. This controller can interact with an identified environment and perform difference task without direct transition and have great advantages over other control methods. IMC is a popular form of interactional based control which is used to regulate the interaction force by robot's impedance. It has been first proposed by Hogan[108]. He showed that both position and force cannot be controlled simultaneously but that the contact force can

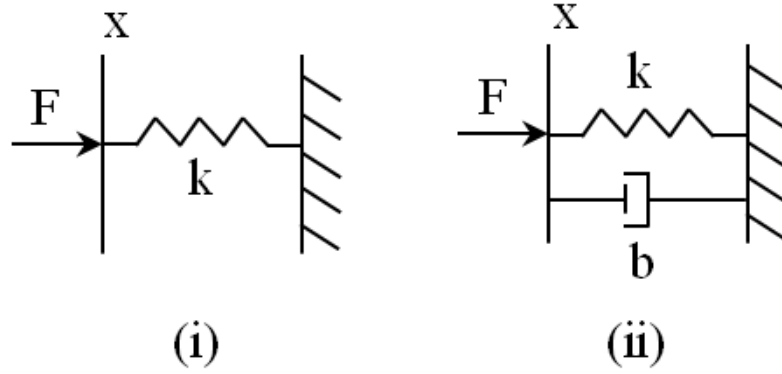


Figure 4.2.: (i) Spring mechanics (ii) Spring Damper mechanics.

be modified by regulating the impedance or compliant of the robot manipulator. For this reason, IMC falls into the category of interactional control. As stated earlier, interactional control focuses on regulating dynamic changes in a robot at an interaction port with environment. The interaction port is defined as the place where exchange of energy occurs. Interactional control develops a relationship between the motion and the force at this port and allows a law to be defined which minimises the difference found from this relationship. In general, interaction method regulates the mechanical impedance or admittance of a robot manipulator. Mechanical impedance refers to a dynamic operator that calculates the force from velocity. Admittance is just the opposite. Common impedance found for different type of mechanical interactions are given in Figure 4.2. The impedance and admittance found from these two interactional model are presented in the Laplace domain in Figure 4.3. In Table 4.1, s is the Laplace domain of motion x .

Table 4.1.: Impedance and admittance function for spring and spring damper model.

	Impedance F/\dot{x}	Admittance \dot{x}/F
Spring mechanics	k/s	s/k
Damper mechanics	b	$1/b$

4.5.1. IMC objective and its structure

In nonlinear robotic system, impedance explains how a manipulator will behave under different environmental interaction. With arbitrary impedance, an arbitrary behaviour of a manipulator can be achieved. The impedance control design is related to find a desired impedance which ensures that the manipulator follows the desired or target impedance. A good impedance behaviour can be achieved with a linear second order model for a manipulator system as shown by Volpe and Khosla [112],[113], which is valid for open and constrained workspaces. The

desired impedance for nonlinear robot dynamics can be selected as,

$$F_c = M_I(\ddot{X}_d - \ddot{X}) + B_I(\dot{X}_d - \dot{X}) + K_I(X_d - X), \quad (4.13)$$

where F_c is the interaction force measured in the contact state, $\ddot{X}_d, \dot{X}_d, X_d$ are the Cartesian desired acceleration, velocity and position, \ddot{X}, \dot{X}, X are the velocity and position of the end-effector. The terms M_I, B_I, K_I are the desired inertia, damping and stiffness parameters (Impedance parameters). Equation (4.13) is chosen to realise compliant robot-object system behaviour as a mass spring damper system with regards to forces exerted by the environment. In the Laplace domain, it is written as,

$$F_c = Z_d(s)(X_d - X), \quad (4.14)$$

where $Z_d = s^2 M_I + s B_I + K_I$. This is similar to the second order linear dynamics of

$$s^2 + 2\zeta\omega_n + \omega_n^2. \quad (4.15)$$

The reason for considering second order dynamics as desired impedance is because of its well known behaviour [114]. The natural frequency ω_n and damping ratio ζ of (4.13) is determined using (4.15) as,

$$\omega_n = \sqrt{\frac{K_I}{M_I}} \quad \zeta = \frac{B_I}{2\sqrt{M_I K_I}}. \quad (4.16)$$

From (4.14) and (4.15), the impedance error is found as,

$$(X_d - X) = \frac{F_c}{Z_d(s)}, \quad (4.17)$$

which is the impedance error ΔX . The impedance control objective is to minimise this error

$$\lim_{t \rightarrow \infty} \Delta X \approx 0 \quad (4.18)$$

by (4.13) with appropriate selection of impedance parameters. If the impedance parameters are selected correctly, then solving (4.13) will provide a commanded position of the end-effector following the contact compliance. The choice of the impedance parameters depends on system impedance behaviour during the contact. It is apparent from (4.16) that, the impedance parameters are related to the natural frequency and the damping of the system. The parameters should be tuned strictly positive which exhibits the critical damping of the system's behaviour. The damping ratio needs to satisfy the following condition:

$$\zeta > 0.5 \left(\sqrt{1 + 2K_r - 1} \right), \quad (4.19)$$

where $K_r = k_e/K_i$ is called stiffness ratio; k_e is the environment stiffness. It depicts that the system's transient properties can be described as a function of environment stiffness and impedance

parameters can be achieved from these properties.

4.5.2. IMC classifications

There are two general impedance control methods available for implementation in constrained manipulator framework based on the fundamental discussion in previous subsection: The position based impedance control and the force based impedance control.

1. Position based impedance controller

In position impedance method (PIM), the mechanical impedance of the manipulator is controlled by generating the desired trajectory of an existing position controller in the joint or workspace. It is widely known for industrial impedance control. Position based impedance law is defined from (4.13) as,

$$F_c = M_I(\ddot{X}_d - \ddot{X}_c) + B_I(\dot{X}_d - \dot{X}_c) - K_I(X_d - X_c), \quad (4.20)$$

where the new term $X_c, \dot{X}_c, \ddot{X}_c$ is called the modified position trajectory. The solution of (4.20) is the impedance error ΔX and X_c can be extracted as,

$$X_c = X_d - \Delta X, \quad (4.21)$$

which is found from the impedance block presented in Figure 4.3. In this method, the desired trajectory is the input to impedance block and the modified output X_c is applied to the inner position control loop. Any position control law as discussed in chapters 3 can be applied to an inner loop which calculates the torque based modified X_c and drives the robot. An impedance block only works when the contact force F_c is measured from the environment i.e. X_d becomes X_r in free space. The environment model is a function of the actual robot end-effector motion and its stiffness. The stiffness is chosen between $10^3 - 10^5 N/m$ to assess the performance of the controller with subject to the interactional force. The desired impedance is achieved if the inner loop ensures precise tracking i.e. $X \approx X_c$.

2. Force based impedance controller

The problem with (4.20) is that it cannot be achieved in general due to the presence of uncertainties. Moreover, the force F_c in (4.20) acts as the desired force input which affects the ability of knowing the stiffness properties of the environment in advance and the desired trajectory cannot be designed to calculate the desired force. Hence, the force impedance method (FIM) is suitable for its force tracking capability with general impedance structure to solve these issues. The changes are made in (4.13) to propose this scheme as,

$$F_d - F_e = M_I(\ddot{\Delta X}) + B_I(\dot{\Delta X}) - K_I(\Delta X), \quad (4.22)$$

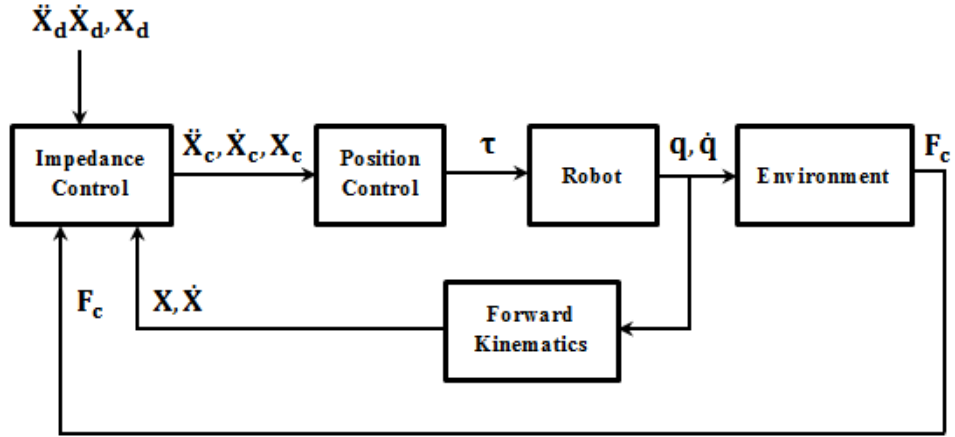


Figure 4.3.: Position based impedance control structure for manipulator.

where $\Delta X = X_d - X_c$ is the impedance error. To derive the above law it is assumed that the robot position control scheme is accurate i.e. $X = X_c$. This allows consideration of the elements of the vectors X and F individually. Consider the environment model discussed in section 4.2 as $f_e = k_e(x - x_e)$. The force error e_f is then found as,

$$e_f = f_d - f_e = f_d - k_e(x - x_e). \quad (4.23)$$

From (4.22) and (4.23), the force error dynamics is found as,

$$m\ddot{e}_f + b\dot{e}_f + (k + k_e)e = k(f_d + k_ex_e) - kk_ex_d, \quad (4.24)$$

where x_d is used to regulate the force error. When x_d is constant, Laplace transformation of (4.24) defines the steady state force tracking error as,

$$e_{ss} = \lim_{s \rightarrow 0} se(s) = \frac{k}{k + k_e} [(f_d + k_ex_e) - kk_ex_d]. \quad (4.25)$$

Equation (4.25) will be zero when the reference trajectory is chosen as,

$$x_d = \frac{f_d}{k_e} + x_e. \quad (4.26)$$

Furthermore, (4.26) only works when the terms x_e and k_e are known *a priori* to generate the position trajectory x_d as a function of k_e . It is assumed that, these variables are known to see the performance of this controller for the Barrett Hand. Then, the adaptive approach will be considered to estimate these variables.

Position Impedance vs Force impedance controller

Two controllers discussed previously use related concepts to achieve the desired impedance

simply by reducing impedance error ΔX to zero. The position-based tactic is less complicated and easier to apply though the method is open loop for the desired impedance. Moreover, the actual robot fingertip position in the present controller might not be achievable for a very soft impedance. Conversely, the force based method ensures tracking with closed loop. Using this method could be computationally affordable in the event that the manipulator is driven at a low speed and considers much less Coriolis and gravity terms. In conclusion, position-based impedance controller is better when it is required to achieve a high position accuracy. The force based method is applicable where the system dynamic effects are negligible.

4.5.3. Contact force analysis and results

Before designing any controller, the contact force needs to be analysed based on the contact mechanics described in section 4.2. Consider one DOF robot link to apply the contact model of (4.5)-(4.9). The model is simulated in MATLAB. The simulation parameters are: Link length, $l = 0.2$, Object mass, $m = 0.25$ and object position, $x_o = 0.3$. A control gain is applied to move the position of the robot link. Control gains are found as, $(k_1, k_2) = (0.9, 0.25)$. The results are presented in Figure 4.4 for different stiffness parameters.

In the figure, the response of joint position θ (top left) is shown before and after contact with the object. The contact is made around 1.1 seconds and the movement of θ slowed down after that. The position of θ is $0.52m$ when stiffness $K_s = 1000N/m$. As soon as the stiffness increased, the value of θ shows less movement after the contact. The position θ stays between $0.56 - 0.57m$ after contact for the stiffness values of $K_s = 1500 - 2500N/m$. The joint torque τ (top left) is smooth and the reaction torque τ_d (bottom left) is zero before the contact. After contact, the robot fingertip exerted by the object force resulting in the high torque values. The values of τ_d also becomes non-zero and increase linearly after balancing the effect of spring mechanics. The bottom right figure shows the contact force. The contact force becomes higher with small amount of stiffness $K_s < 10^3 N/m$. Now, consider the constraint Barrett hand dynamics in joint space as,

$$M_f \ddot{q} + C_f(q, \dot{q}) + G_f(q) = \tau - \tau_d \quad (4.27)$$

$$= \tau - J^T F_c, \quad (4.28)$$

where τ defines the driving torque and $J^T F_c$ translates the contact force to the joint space as reaction torque. If the inverse kinematics have a specific solution, simulation takes less time in joint space and workspace coordinates are easily found from joint positions. The contact model between the robot and the object is developed in SimMechanics. The simulation structure for contact force measurement is given in Figure 4.5. The Barrett finger F_3 is chosen to carry out the simulation. The CTC based controller is chosen to command the fingertip trajectory in workspace. The CTC based parameters are found by trial and error as $K_p = 430$, $K_v = 175$.

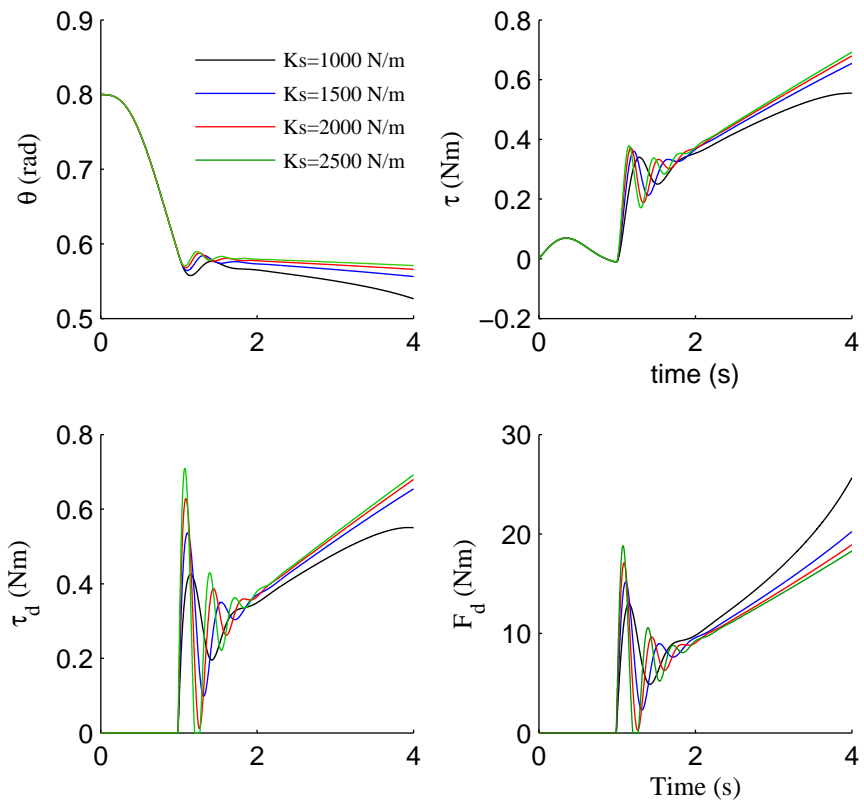


Figure 4.4.: Position, force and torque results of one link planar finger during the interaction with an object.

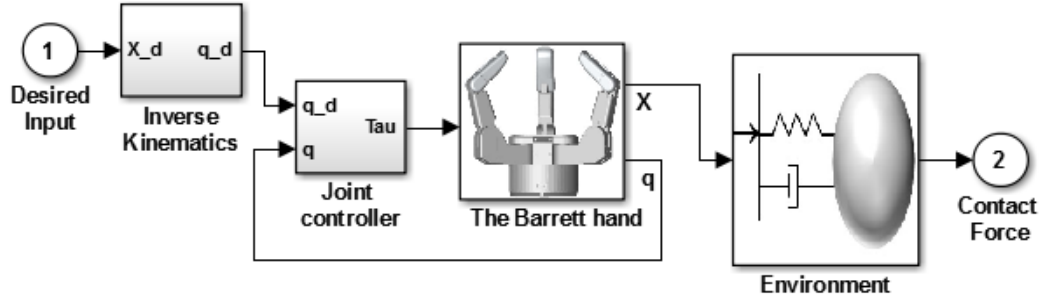


Figure 4.5.: Contact force measurement model of Barrett hand.

The contact location, $x_c = 0.035\text{ m}$ is considered to keep the object within workspace range of the hand.

Figure 4.6 illustrated the contact force measurement for different stiffness starting from $10 - 50\text{ kN/m}$. The simulation shows that the contact force is higher for a very rigid (50 kN/m) object. The force reaches up to 250 N which is large enough to damage the object. The torque generated from the object will be also high which will cause damage to the joint actuator. The amplitude of the force signal is also large which indicates vibration between the finger and the object. As stiffness goes down, the force becomes lower. The reaction torques of both finger joints found from the generalised contact forces are presented in Figure 4.7. The results show that the torques are increasing with respect to an increased stiffness of the environment. It is necessary to see the condition of the fingertip motion before and after contact to analyse the transition behaviour.

Figure 4.8 is obtained with the fingertip trajectory and the velocities of the Barrett fingertip F_3 . The trajectory started with an initial condition of $z_o, y_o = (0.1565, 0.1122)\text{ m}$. The actual position (blue line) is not converging to the desired position. The contact is made around 2.1 seconds of the simulation and the motion is found to be oscillating due to exerted force on the fingertip. This happens with velocities which are not converging to zero. After contact, the velocities are oscillating with a low amplitude which suggests that the finger is penetrating the object and cannot reach further. In Figure 4.9, the contact force simulated in the workspace is presented for comparison. In this case, the stiffness, $K_e = 30\text{ kN/m}$ is used.

The correct interactional model provides the force only after the contact is made between the fingertip and the object. In practical situation, the fingertip does not cross the object. This implies that the force after the contact and the actual fingertip position must be within the range of the robot-object workspace coordinates for correct interactional model. The results of the Figure 4.6-4.9 indicate the authenticity of the interactional model. Figure 4.6 is obtained with the contact force occurred due to the interaction between F_3 and the object. For stiffness of $k_s = 10 - 50\text{ kN/m}$, the contact forces are found between $110 \sim 240\text{ N}$ with oscillation. The oscillations are occurred due to the joint damping and the object stiffness. In Figure 4.7, reac-

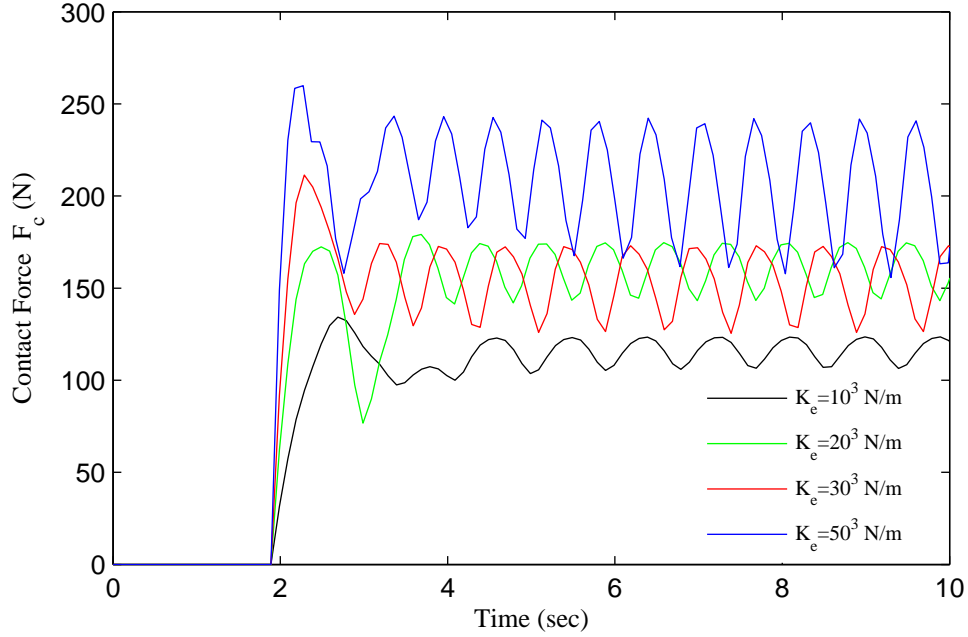


Figure 4.6.: Contact force F_c calculation of Barrett Finger F_3 for different object stiffness K_e values.

tion torques τ_d are shown for the same stiffness values. The torques are also achieved between $-10 \sim 16 Nm$. The contact forces and the reaction torques do not exhibit any sudden changes in their responses. Figure 4.8 displays the position and velocities of the finger during the contact. The actual position (blue line) of the top figure stays around $0.035 m$ with oscillation after the contact transition (3 seconds). This indicates that the fingertip tries to penetrate the object during contact but do not cross over. The fingertip crossing over the object may result in drastic changes in the response and prove the interactional model to be wrong.

The advantage of the interactional model is that it allows to see the manipulator response, in contact with the object for various object stiffness values. It is apparent from Figure 4.4 and Figure 4.6 that, the behaviour of the contact force is not similar between one link planar and the Barrett finger. The force increases linearly in time for planar finger. The Barrett finger shows the contact forces stays between certain values ($-10 \sim 16 Nm$) for an additional DOF. The extra DOF splits the energy of the system and the fingertip penetrates the object with less energy during the contact. Hence, when the contact is made, forces rise vertically but settle down with oscillations. It happens due to the actuator damping which limits the joint to push the object deliberately without any constraint.

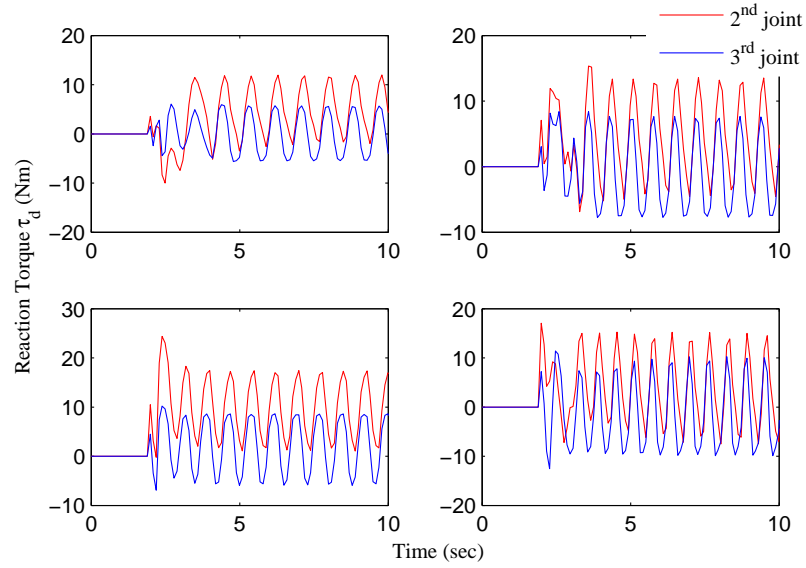


Figure 4.7.: Reaction torque τ_d of the finger F_3 joints. Top left $K_s = 10^3 \text{ N/m}$, Top right $K_s = 20 \text{ kN/m}$, Bottom left $K_s = 30 \text{ kN/m}$, Bottom right $K_s = 50 \text{ kN/m}$.

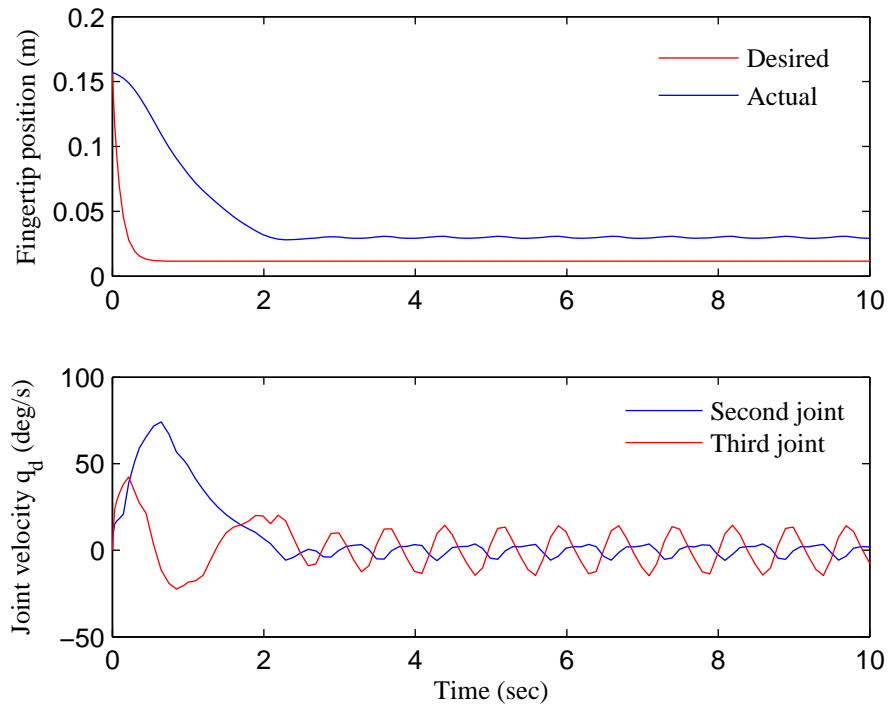


Figure 4.8.: Fingertip motion of F_3 (Top) and joint velocities \dot{q}_i (Bottom).

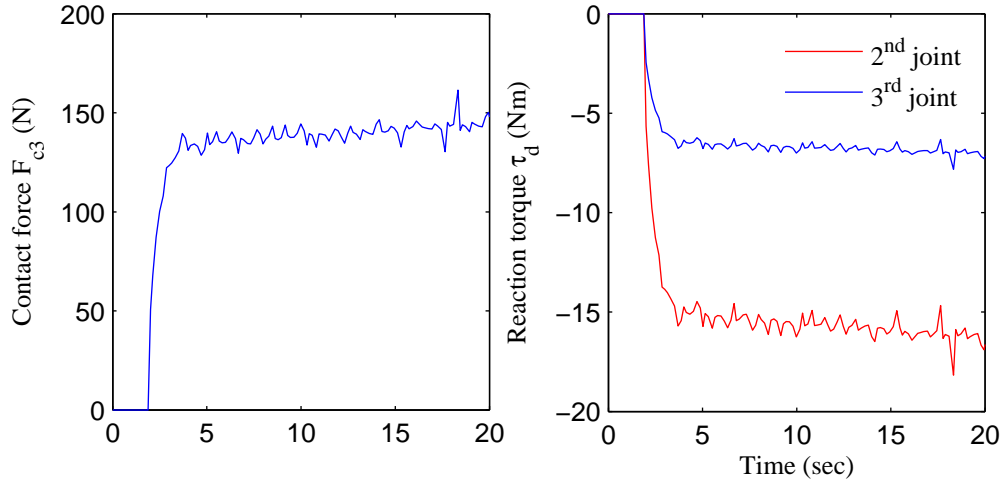


Figure 4.9.: Contact force and reaction torque measured in workspace based control of finger F_3 .

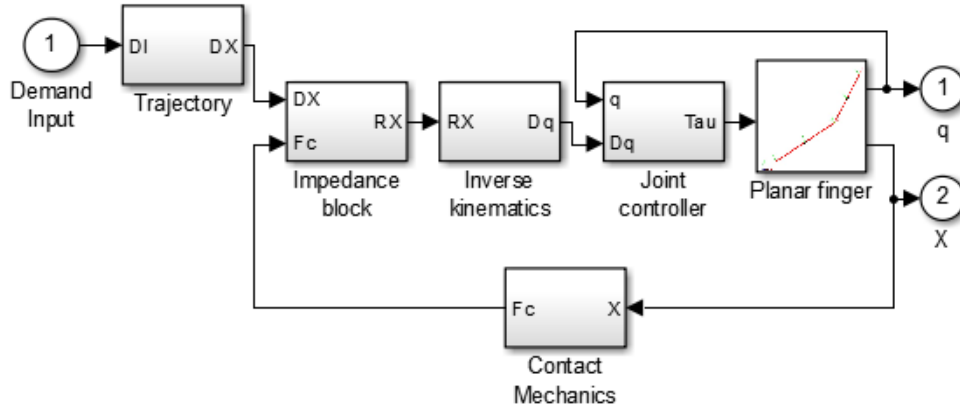


Figure 4.10.: Position based impedance control schematics of planar finger.

4.5.4. Planar finger control with PIM

In this section, PIM control method is applied to a planar finger. The control structure is shown in Figure 4.10.

The finger is commanded to track the Cartesian position trajectory of

$$dx(t) = dx_i e^{-at} + dx_i (1 - e^{-at}), \quad (4.29)$$

where dx_i is the desired cartesian position to be reached by the finger. When the planar finger reaches the object position, the model defined in (4.5) is activated. The impedance block is driven by (4.20). The output of the impedance block is the impedance error e_I that is applied to

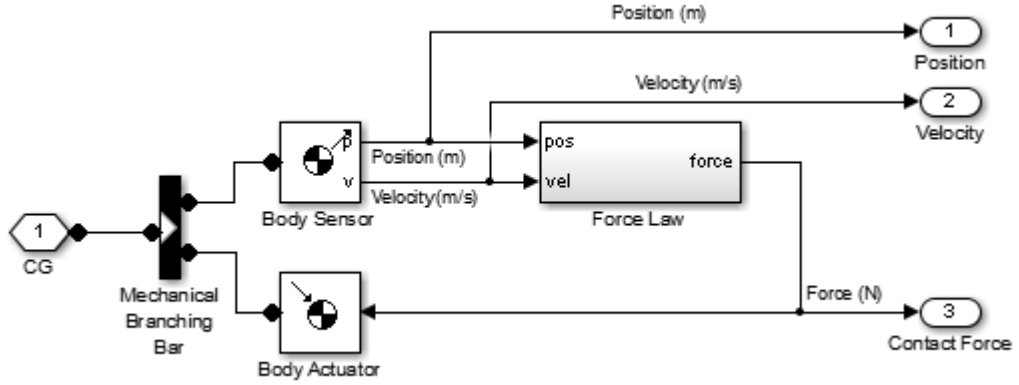


Figure 4.11.: SimMechanics contact model between robot link and object.

get the modified trajectory x_c . The desired joint values q_d are found through x_c . When no forces are sensed, the trajectory relation becomes $x_d = x_c$ and general position control is applied in the joint space. The trajectory is modified according to the measured contact force value and to behave as mass spring damper based impedance filter of (4.20). According to the figure, the joint controller is formed in a separate loop and not related to the impedance block. The model based joint controller is available to track the desired joints calculated from x_c . The finger is developed in SimMechanics which provided joint and fingertip motion through sensors. The sensor values are fed back to the impedance and joint controller block.

Contact mechanics model development was not straightforward in the SimMechanics toolbox as this is not just the equation derived in the contact mechanics section. The model should exhibit the realistic behaviour for contact collision between two objects. The model developed in SimMechanics is given in Figure 4.11. Channel 1 is connected to the second link body block (not visible). The other side has two parts: sensor and actuator. The sensor passes the trajectory data to the force law block and the measured force exerts on the fingertip by body actuator block. All links are interconnected to ensure that no slip occurs during the contact state. In physical case, contact forces are only found when two objects are in contact and considered as coupled system. The same technique is used in the SimMechanics development. The simulation parameters are: link mass, $(m_1, m_2) = (1, 1) \text{ kg}$, link length, $l_1, l_2 = (0.5, 0.5) \text{ m}$, joint control gain, $k_p, k_v = (80, 37)$, object location, $x_e = 1000 \text{ N}$, trajectory constant, $a = 1$, and impedance parameters are chosen based on trial and error as, $M_I = 10 \text{ kg}$, $B_I = 132 \text{ N s/m}$, $K_I = 500 \text{ N/m}$.

Figure 4.12 presents the results of the PIM control of a planar link. The top left figure provides the angles in radian. The top right one shows that the velocity tends to zero with quick settling (0.1 second). This means that the system is stable as system dissipates its energy to reach to the steady state. The middle left figure shows desired, commanded and actual trajectory of the fingertip. When the contact is made and the impedance filter becomes active in the closed loop,

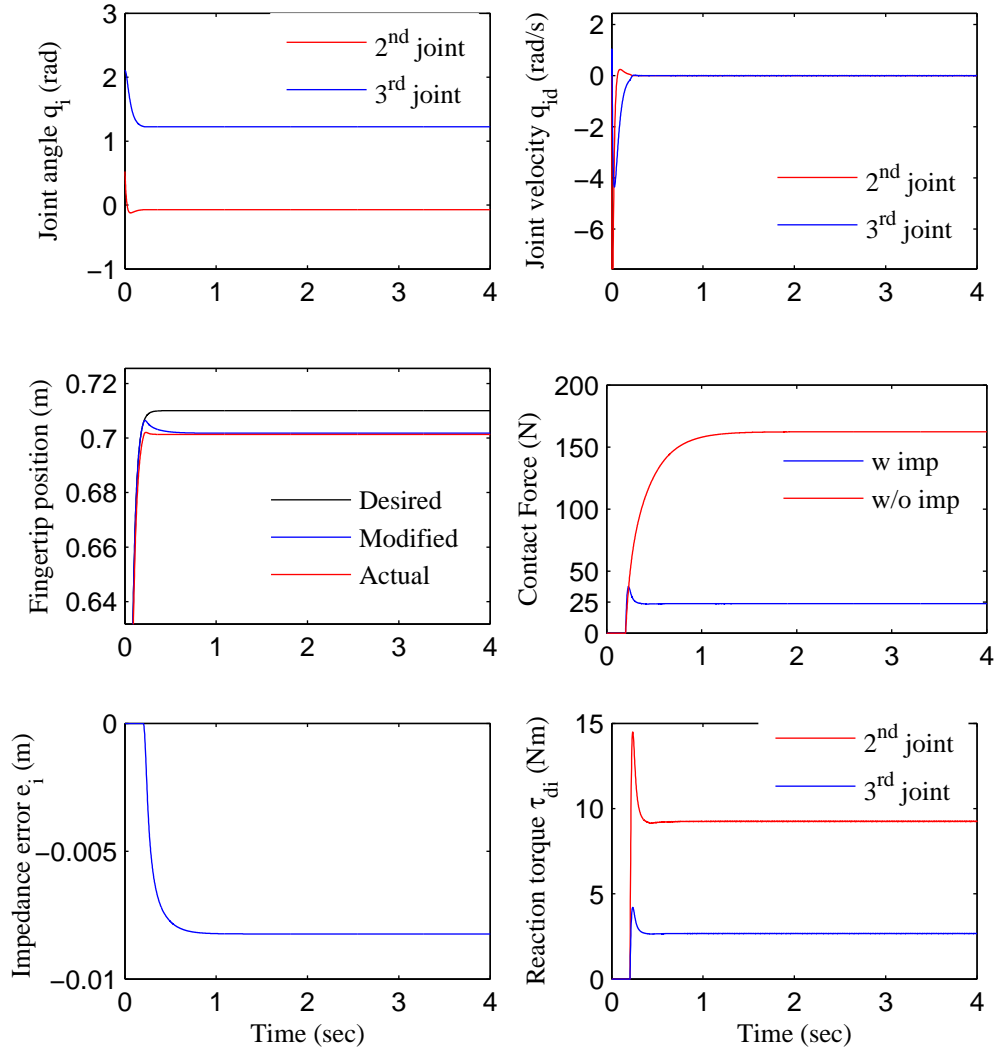


Figure 4.12.: Position based impedance control results from planar finger.

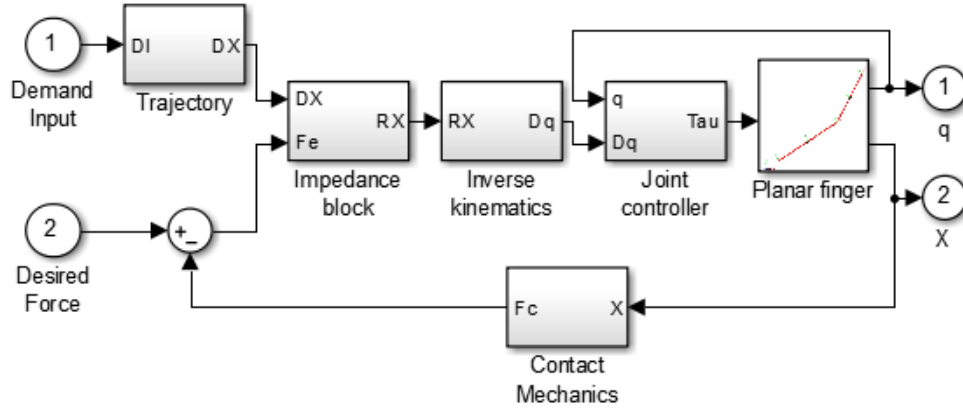


Figure 4.13.: Force based impedance control diagram for planar finger.

the modified trajectory rx (blue line) changes its path from the desired trajectory dx (black line). The actual trajectory x first follows dx , then tracks rx . This indicates the theoretical validation of $dx = rx$ before contact. The middle right figure is presented with the contact force before and after impedance. The force goes above $150N$ without impedance control mode. The contact force settles down at $25N$ with the desired impedance of the system. The bottom two figures show the impedance error ΔX and reaction joint torque τ_{di} results respectively. The impedance control proved to be very efficient in keeping the force down at optimum level. It would be more beneficial if the contact force can be controlled arbitrarily. This is discussed in the next subsection.

4.5.5. Planar finger control with FIM

In this section, the planar finger is simulated in the constrained environment. The control of all parameters remains the same as in the previous section except for the impedance parameters. The impedance parameters are found with trial and error as, $M_I = 8.9kg$, $B_I = 167Ns/m$ and $K_I = 560N/m$. The desired force is considered as $F_d = 10N$. The control structure is given in Figure 4.13. The first difference in this control structure is the presence of force error e_f calculated from the desired F_d and the actual F_c force. The error e_f drives the impedance filter equation (4.20) to achieve modified position trajectory. The second difference is the modification of position trajectory in constrained direction as,

$$dx = x_e + f_d/k_e, \quad (4.30)$$

where dx is generated based on the contact force f_d . The selection of (4.30) approves the steady state force error equation of (4.25) to zero.

The results of FIM methods for planar link are presented in Figure 4.14. The middle right figure

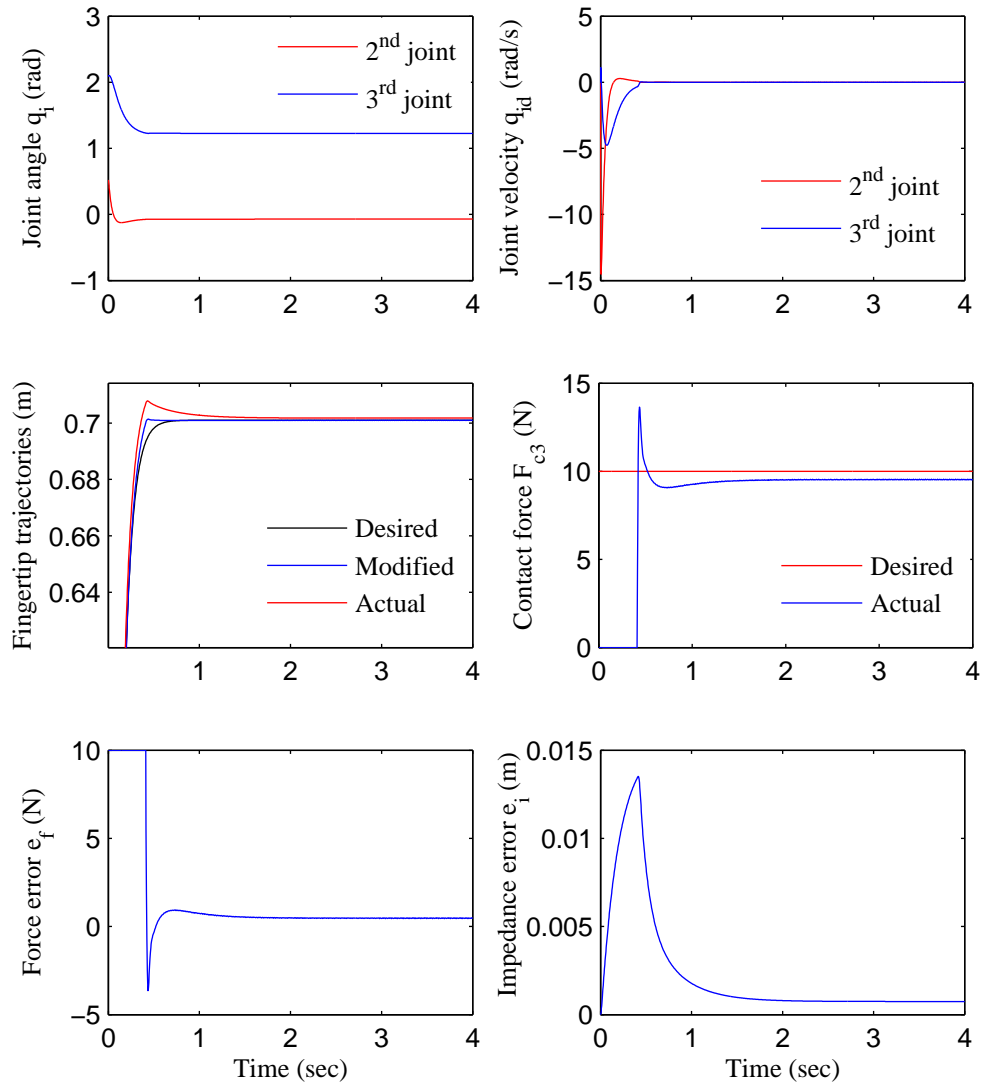


Figure 4.14.: Force based impedance control results from planar finger.

shows the contact force for desired 10 N (Red line). The bottom left and right figure shows the force and impedance error approaching to zero in time respectively. The advantage of a force based mode is the selection of an arbitrary contact force to be tracked by the interaction force. This is shown in the middle right figure for a given desired force, $F_d = 10\text{ N}$. Arbitrary contact force tracking cannot be achieved for position based impedance control (results of Figure 4.12). The problem in FIM mode is that the environment location x_e and the environment stiffness k_e need to be known in advance in order to achieve successful tracking. Environment parameters estimations are proposed in section 5.5.8.

4.5.6. Position and force impedance control of Barrett finger

The control structure for PIM implementation to the Barrett hand is displayed in Figure 4.13. The SimMechanics based Barrett finger F_3 is used in this case to see the performance of the controller. The mass and link dimensions are considered from chapter 2.2. The trajectory constant, $a = 0.01$, joint controller gain, $K_p, K_V = (430, 165)$, object location, $z_o = 0.035\text{ m}$ are employed. The finger is commanded to reach a position greater than 0.03 m in the z direction. The environment stiffness is chosen as $k_e = 10^3\text{ N/m}$. The impedance parameters are found as $M_I = 15\text{ kg}$, $B_I = 780\text{ Ns/m}$ and $K_I = 5000\text{ N/m}$.

The results are presented in the Figure 4.15. The top two plots present the joint angle trajectory position and the velocity of F_3 . The middle left figure shows desired, actual and commanded fingertip trajectory of the finger. The tracking error is visible but actual fingertip trajectory is quickly tracking the modified trajectory. The contact force is shown on the middle right. The impedance error is very small and settling down which means the modified position is consistent after the contact. Force behaviour for different stiffness values are presented in Figure 4.16. For stiffness values between $10 - 50\text{ kN/m}$, the force settled down between $20 - 30\text{ N}$. When the stiffness is 10^3 N/m , the force settled down around 17 seconds with large oscillations. The peak and the steady state response are 58 N and 27 N respectively. The difference between them are 21 N . The settling down time is 5.1 seconds for higher stiffness, $k_e = 50\text{ kN/m}$. In this case, the difference between peak (24 N) and steady state response (20 N) are only 4 N . It is noticeable that, force response is achieved with less settling time when robot fingertip interacts with hard (high stiffness values) object. In soft (low stiffness values) interaction case, the force settles down but with long settling time and oscillations. According to the results obtained, the impedance controller minimised the force for different stiffness values. However, the impedance controller shows better performance with hard object interaction. The contact force performance based on FIM is given in Figure 4.17. In the Figure, FIM is applied to all fingers of the Barrett hand with desired force of 15 N . Spread angle is considered in this case for finger F_1 and F_2 . All contact forces are converged very close to the desired force value.

The PIM and FIM control are applied to both planar and Barrett finger. In both cases, contact forces have been optimised. The advantages of FIM based mode is that the independent choice

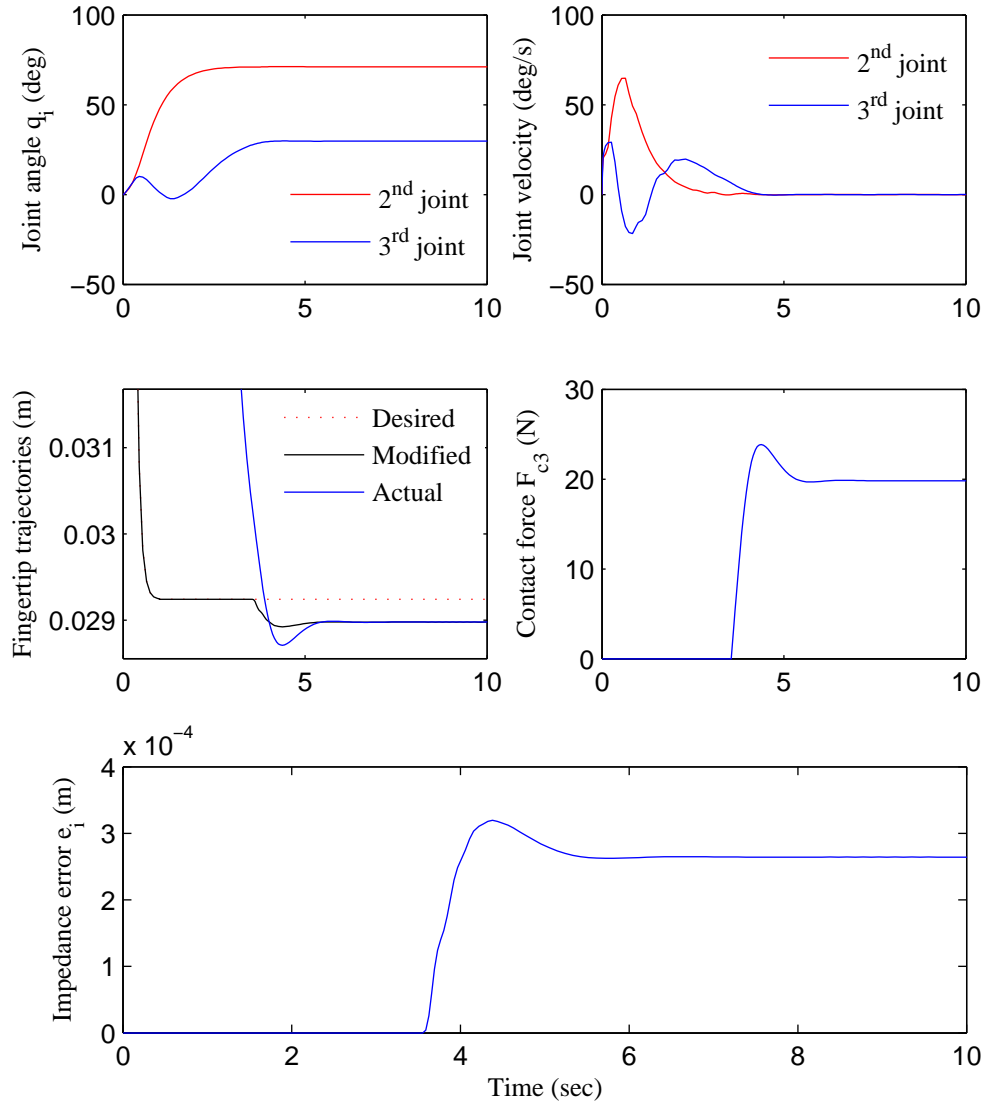


Figure 4.15.: Position Impedance control of Barrett finger F_3 , $K_e = 10^3 \text{ N/m}$.

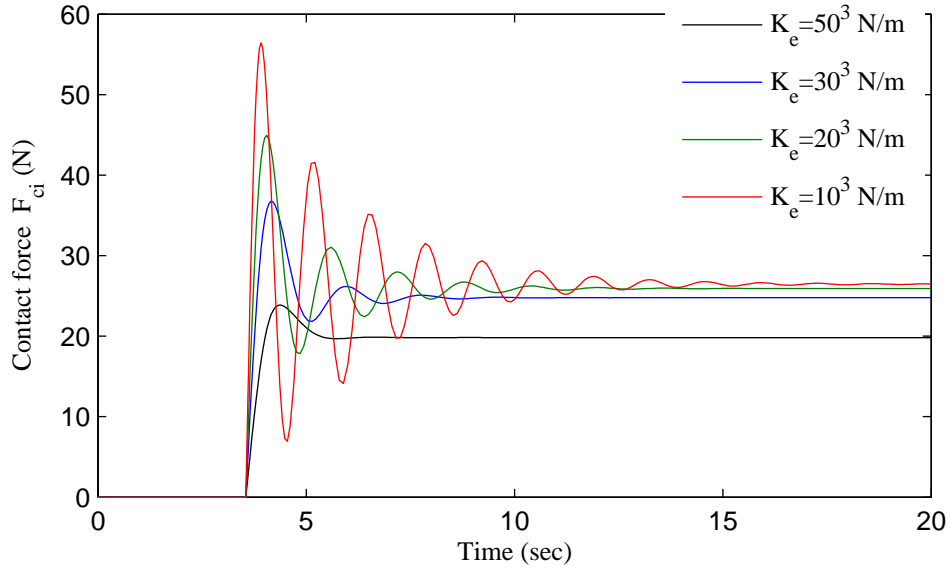


Figure 4.16.: Position based impedance control of the Barrett finger F_3 for different stiffness.

of the desired force and the ability of being tracked by the regulated interactional force. In grasping and manipulation, an arbitrary selection of interactional force would be an important feature to accomplish the tasks.

4.6. Adaptive force impedance control

As stated in the section 4.2, the environment location and corresponding properties are pre-requisite to achieve force tracking characteristics with impedance control. From a practical viewpoint, these information are not obtained from the controller in advance. The unavailability of the information greatly affects the online based position trajectory of (4.30). This trajectory depends on the object location x_e and the stiffness k_e . The unknown variation of these properties as $\Delta x_e, \Delta k_e$ can be considered as follows,

$$x_{est} = x_e + \Delta x_e \quad k_{est} = k_e + \Delta k_e, \quad (4.31)$$

where x_{est} and k_{est} are the estimation of the variables. Considering these variations, steady state force error of the closed loop can be written as,

$$e_{ss} = \frac{k}{k + k_e + \Delta k_e} \left[k_e \Delta x_e - \frac{\Delta k_e}{k_e} f_d + \Delta x_e \Delta k_e \right]. \quad (4.32)$$

In general, the value of k_e is large. Therefore, a very little variation of Δx_e may induce a large force error. To eliminate this problem, adaptive schemes are proposed considered from previous findings [109],[110]. The objective of this adaptive based controller is to estimate

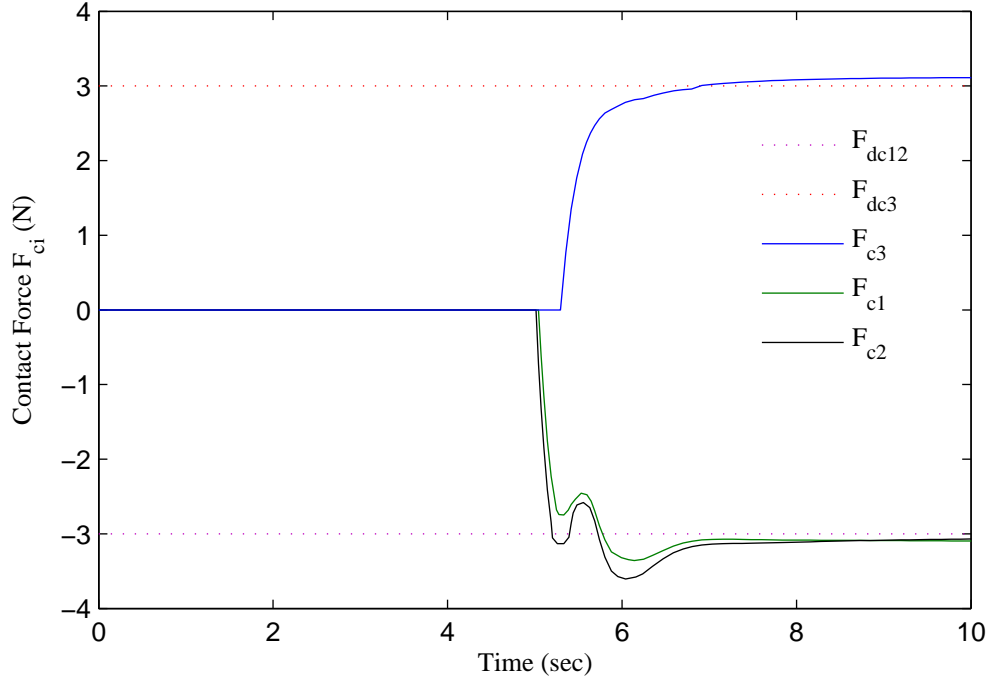


Figure 4.17.: Force based impedance control of Barrett finger for desired force $F_d = 3N$, stiffness value, $k_e = 10^3 N/m$.

the object stiffness k_e and location x_e online and calculate the reference position x_d in the constrained direction.

It is possible to define x_d in terms of adaptation as,

$$\hat{x}_d = \hat{x}_e + \frac{1}{\hat{k}_e} f_d, \quad (4.33)$$

where \hat{x}_e and \hat{k}_e are adaptation based estimation of x_e and k_e respectively. Now, the generalised contact force from (4.5) after contact can be written as,

$$f_c = k_e(x - x_e). \quad (4.34)$$

The form in (4.34) suggests with adaptation as,

$$\hat{f}_c = \hat{k}_e(x - \hat{x}_e), \quad (4.35)$$

where \hat{f}_c is the current estimate of the interactional force based on \hat{x}_e and \hat{k}_e . Lets consider the variation parameters are, $\phi_k = \hat{k}_e - k_e$, $\phi_x = \hat{k}_x - k_x$, $\phi = [\phi_k \quad \phi_x]^T$ and by subtracting (4.34)

from (4.35), the following can be found,

$$\hat{f} - f_c = [x \quad -1]\phi. \quad (4.36)$$

Therefore, the adaptive FIM problem can be stated as a scheme which will adjust parameters \hat{k}_e, \hat{k}_x according to (4.36) which will converge $\hat{f} \rightarrow f_c$ as $t \rightarrow \infty$. The update laws for \hat{k}_e, \hat{x}_e can be derived using Lyapunov based method which ensures $\hat{f} \rightarrow f$ means $f \rightarrow f_d$. The Lyapunov candidate can be defined as,

$$V = \phi^T \Gamma \phi, \quad (4.37)$$

where Γ is positive definite matrix. The update law is declared as,

$$\dot{\phi} = -\Gamma^{-1} \begin{bmatrix} x \\ -1 \end{bmatrix} (\hat{f} - f_d). \quad (4.38)$$

Differentiating (4.34) along (4.37) gives,

$$\dot{V} = 2\phi^T \Gamma \dot{\phi} = -2\phi^T \begin{bmatrix} x \\ -1 \end{bmatrix} (\hat{f} - f_d) = -2(\hat{f} - f_d)^2. \quad (4.39)$$

If ϕ is adjusted by following (4.38), then (4.37) and (4.39) indicates that $\hat{f} \rightarrow f$ as $t \rightarrow \infty$. The adaptation law for k_e and x_e are obtained as,

$$\begin{aligned} \dot{\hat{k}}_e &= -\gamma_1 x (\hat{f} - f) \\ \dot{\hat{x}}_e &= \frac{(\hat{f} - f)}{\hat{k}_e} (\gamma_1 x \hat{x}_e + \gamma_2) \end{aligned} \quad (4.40)$$

where γ_1, γ_2 are positive constants. All adaptation based control laws are combined together below,

$$\begin{aligned} x_d &= \hat{x}_e + \frac{1}{\hat{k}_e} f_d \\ \dot{\hat{k}}_e(t) &= \hat{k}_e(0) - \gamma_1 \int_0^t x (\hat{f} - f) dt \\ \dot{\hat{x}}_e(t) &= \hat{x}_e(0) + \gamma_1 \int_0^t \frac{(\hat{f} - f)}{\hat{k}_e} (x \hat{x}_e + \gamma_2 / \gamma_1) dt \\ \dot{\hat{f}} &= \hat{k}_e (x - \hat{x}_e) \end{aligned} \quad (4.41)$$

4.6.1. Adaptive force impedance method of planar finger

The adaptive control laws of (4.41) are applied to the planar finger. The same parameters used for the planar FIM is used in this case. The adaptation parameters are: $\gamma_1 = 1$ and $\gamma_2 = 0.5$. The desired contact force is, $f_d = 30N$. The results are shown in Figure 4.18. The top two figures show the contact force and the force error. The force is converged to the desired values. The overshoot in the force response is small in comparison to the contact forces derived without adaptation as in Figure 4.13 and 4.14. The force and impedance error approached zero within

a short time frame. The initial condition of $\hat{k}_e(0)$ considered as 400 and estimated around 409 N/m . The estimated force \hat{f} is also found around 30 N . The online desired trajectory x_r settled down at 0.7 m through adaptation which is the fixed position of the environment.

4.6.2. Adaptive force impedance method of Barrett finger

The adaptive control laws of (4.41) are also applied to the Barrett finger F_3 . The trajectory constant, joint controller values and other parameters are applied from section 4.5.6. The adaptation parameters are: $\gamma_1 = 0.0018$, and $\gamma_2 = 0.001$ chosen by trial and error. Desired contact force is selected as 10 N . The results are given in Figure 4.19. The contact (top left) is made at 6.5 seconds and the contact force is settled down around 7 seconds. The peak force is measured about 12 N before reach the steady state value of 10 N . The top right figure shows the force steady state error. The error F_e approaches closure to zero around 7 sec with final value of $\sim 0.02\text{ N}$ (2% error). The impedance error is also found to be very close to zero ($2.5 \times 10^{-3}\text{ m}$). This implies that the modified trajectory obtained through the impedance filter is well tracked by the actual fingertip trajectory after the contact. The object stiffness value is estimated online as $\sim 4 \times 10^3\text{ N}$. The estimated force by adaptation is obtained as $\sim 7\text{ N}$. The bottom plot is presented with online trajectory of the fingertip F_3 which was calculated from \hat{k}_e and \hat{x}_e . The adaptive based FIM model suggests the excellent force tracking ability of the interactional force. For both planar and Barrett finger, the contact forces results are better compared to general FIM mode. The online trajectory x_r is estimated accurately at the object location after the contact .

4.7. Evolutionary algorithm based impedance control

In sections 4.5 and 4.6, the impedance and the adaptive impedance based control methods are discussed. These methods are based on the tuning of impedance parameters by trial and error. This tuning process is time consuming and does not give any indication where to start with the initial selection of the parameters. Evolution based techniques are recent approach to obtain the impedance values [115],[116]. This is an optimisation technique based on the collective learning process. A set of population candidate is chosen, each of which searches for potential solution in the space of a given problem. The population is arbitrarily selected, and it evolves toward better neighbourhood of the search space.

4.7.1. Evolutionary algorithm

Evolutionary algorithm (EA) searches for the optimal weight values w_j of a system network. According to the optimal control, the objective is to find the controller which provides the best

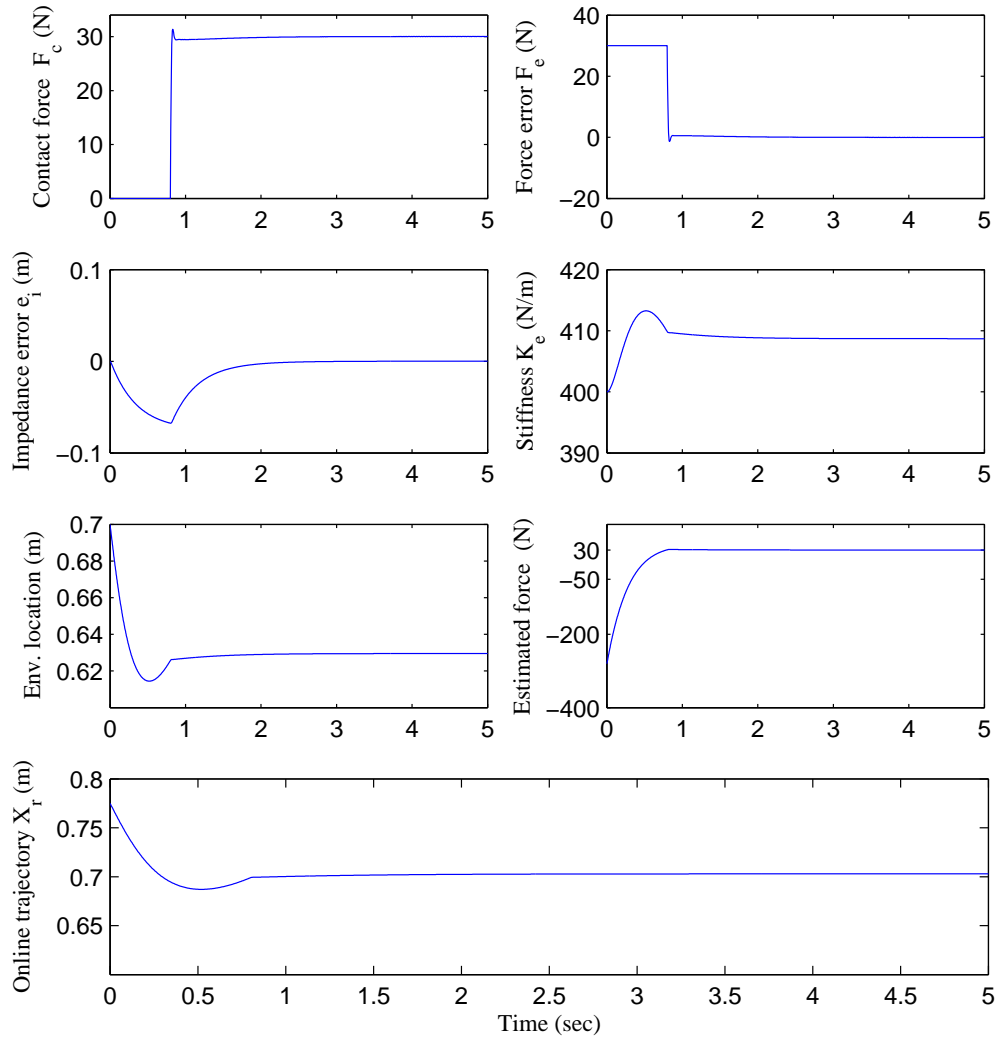


Figure 4.18.: Adaptive impedance control of planar fingers for desired force $F_d = 30\text{ N}$.

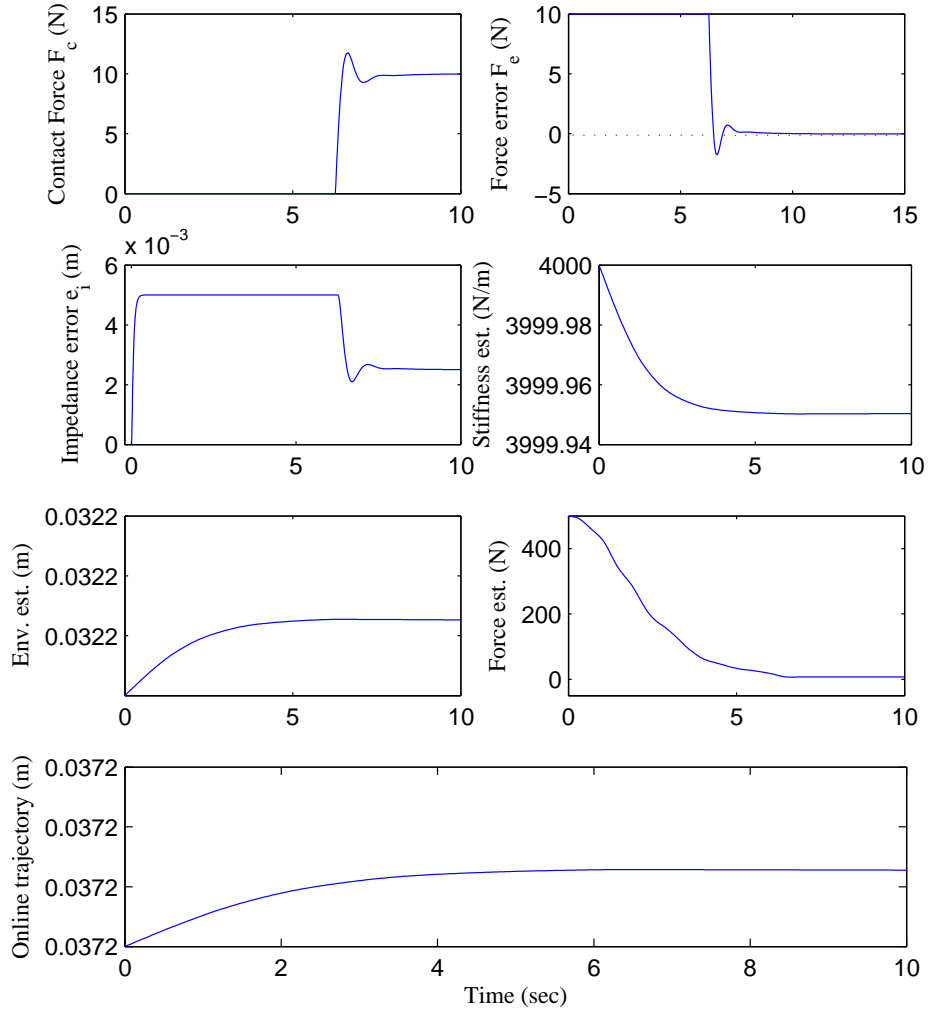


Figure 4.19.: Adaptive impedance control of Barrett finger for desired force $F_d = 10\text{N}$.

performance. Whether the method is classical optimal or evolution, there must be an optimisation procedure exists. The optimisation procedure is inspired by Darwinian natural selection concept [117]. There are different methods available for applying EA but all are formed based on the natural selection principle and processed by mutation, crossover or recombination. The process continues until the fitness value of each weight is determined. The populations are generated iteratively to achieve better solutions.

The evolutionary process used in neuroevolution is commonly based on Genetic algorithm (GA) or Evolution strategy (ES). GA is implemented with binary vectors where the ES method is applied by real scale vectors to determine the solution. Crossover and mutation searching process is common in GA whereas ES method is applied with recombination and mutation based searching. The another advantage of ES is the implementation ability in continuous domains. These make ES mode suitable for continuous domain optimisation problem and real time application. These all are discussed in [118, 119, 120].

The general impedance controller structure is similar to the second order system. In that case, classical control theory is used to determine the impedance parameters on the basis of critical damping factor and natural frequency of the closed loop system. However, the selection of accurate frequency or damping for an environment is not an easy task. Moreover, the contact stability should be compromised as not even a single oscillation might not be allowed in industrial contact environment. Evolutionary technique can be used in order to provide automatic procedure for optimal selection of impedance parameters considering contact stability.

In this section, evolution based optimisation is implemented to determine the impedance parameters for constrained robot-environment framework. The impedance controller derived previously in this chapter is established under Evolutionary algorithm. The algorithm is used to optimise the impedance parameters. One of the popular evolution based ES technique is the Covariance Matrix Adaptation - Evolution strategy (CMA-ES). This is applied to achieve the optimisation within the problem objective rather than manually chosen impedance parameters [121]. It adapts the covariance matrix at each step according to the fitness values of the current population instead of random selection of the strategy parameters. Each individual population candidate is expressed as real valued solution vectors which are processed by recombination and mutation. In mutation, normally distributed random vector is added with zero mean and the covariance matrix of the distribution is adapted to boost up the search procedure. The standard deviation of the mutation distribution is called sigma which indicates the convergence of the evolution process. Mutation is understood by adding a normally distributed random vector with zero mean, where the covariance matrix of the distribution is itself adapted during evolution to improve the search strategy.

4.7.2. Discretized impedance controller

The first criteria to find optimal parameters based on the CMA-ES based technique is to define the fitness function. The fitness function is formed such that it minimise the force error between the desired and the actual force and also ensure the contact stability between the robot and the environment. The fitness function is then applied to CMA-ES algorithm which provides the optimal values of impedance parameters M_I, B_I, K_I of (4.22). Let's represent (4.22) in Laplace domain as,

$$H(s) = \frac{E(s)}{F(s)} = \frac{1}{M_I s^2 + B_I s + K_I}, \quad (4.42)$$

where $E(s)$ is the position error between desired and regulated position. The term $F(s)$ is the force error in the Laplace domain. The second order characteristic equation is equal to (4.42) as,

$$H(s) = \frac{1}{M_I s^2 + B_I s + K_I} = \frac{1}{s^2 + 2\zeta\omega_n s + \omega_n^2}, \quad (4.43)$$

where ζ and ω_n are the damping and natural frequency of the system. From (4.43) the following relationships are achieved,

$$\begin{aligned} \omega_n &= \sqrt{\frac{K_I}{M_I}} \\ \zeta &= \frac{B_I}{2\sqrt{K_I M_I}} \end{aligned} \quad (4.44)$$

It is clear from (4.44) that the system parameters depends on impedance parameters. The critical damping value should be less than one for second order system to ensure response without overshooting. Also, the frequency ω_n should be lower than the internal position controller frequency. It is not simple to select M_I, B_I, K_I on the basis of (4.44). The impedance filter (4.42) is discretized to apply the Evolutionary method. The bi-linear transformation of $H(s)$ is expressed as,

$$H(z) = H(s) \Big|_{s=\frac{2}{T} \frac{z-1}{z+1}}. \quad (4.45)$$

From (4.42) and (4.45),

$$H(z) = \frac{E(z)}{F(z)} = \frac{T^2(z+1)^2}{w_1 z^2 + w_2 z + w_3}, \quad (4.46)$$

where

$$\begin{aligned} w_1 &= 4M_I + 2B_I T + K_I T^2 \\ w_2 &= 2K_I T^2 - 8M_I \\ w_3 &= 4M_I + K_I T^2 - 2B_I T \end{aligned} \quad (4.47)$$

From (4.46) and (4.47), the digital differential equation is implemented as,

$$E(n) = \frac{1}{w_1} (T^2 F(n) + 2T^2 F(n-1) + T^2 F(n-2) - w_2 E(n-1) - w_3 E(n-2)). \quad (4.48)$$

The equation (4.48) is the expression of the discretized impedance filter where F and E based terms on the right side are input to the filter, T and w are the weighting values and $E(n)$ is the

single output. The CMA-ES algorithm is used to search the optimal values of M_I, B_I, K_I and the weighting values are computed using (4.49). The fitness function is defined to minimise the force error as,

$$f_t = \frac{\sum_{k=1}^N |f_d - f_k|}{N}, \quad (4.49)$$

where f_d is the desired force and the term f_k is the current force at time step k . The sample is expressed as N . Damping ratio condition of (4.19) is considered in this to achieve the critical damping at the force output. The impedance block of Figure 4.13 is replaced to establish the CMA-ES algorithm based impedance controllers as given in Figure 4.20.

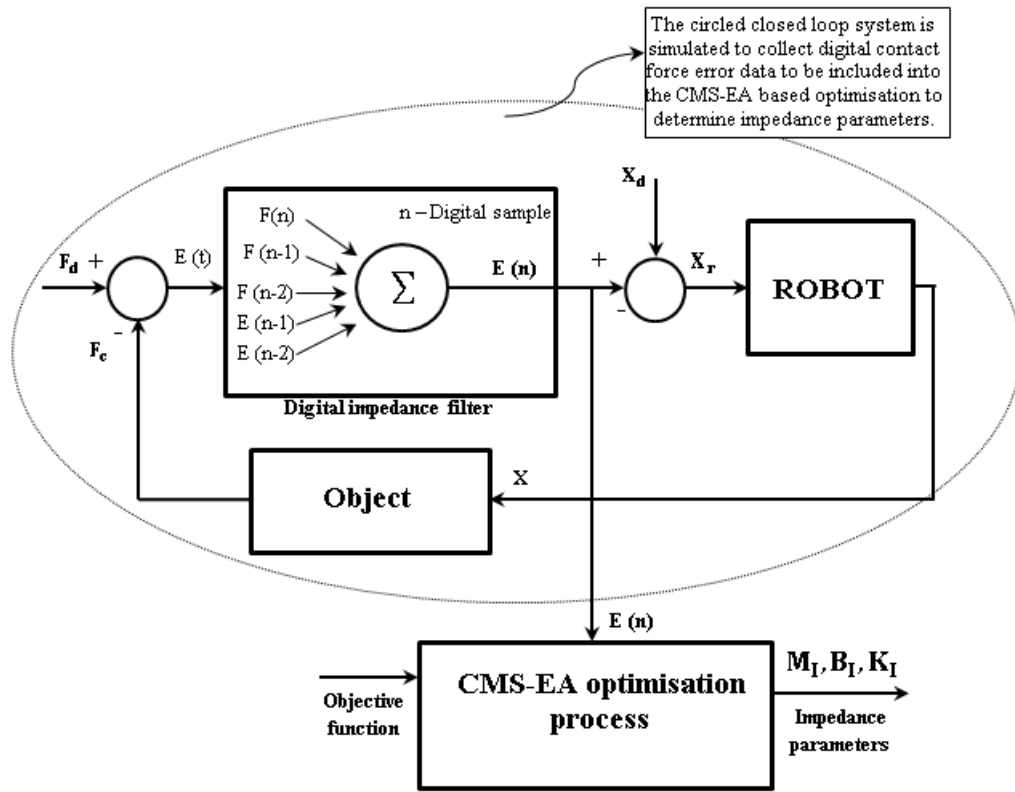


Figure 4.20.: CMS-EA based impedance parameter selection process from digital impedance controller based robot-object interactional environment.

In the circled section of Figure 4.20, the digital impedance filter output $E(n)$ is calculated according to the discrete differential equation of (4.50). The modified trajectory X_r is calculated from the difference of desired trajectory X_d and the digital force error $E(n)$ when the robot fingertip interacts with the object. The force error $E(n)$ is applied to the CMS-EA based algorithm along with the objective function stated in the equation (4.51) (Bottom of Figure 4.20). The impedance parameters M_I, B_I, K_I are obtained from the optimisation process. The parameters are then applied to the Force based impedance control presented in the Figure 4.13 during the

interaction between fingertip-object. In this case, the Barrett finger is used instead of planar finger.

4.7.3. Simulation procedure

The CMS-EA based impedance controller is an extended version of the force based impedance controller applied to the Barrett Hand. The digital impedance filter block of Figure 4.20 transforms the analogue impedance filter. Then CMS-EA algorithm was used in the MATLAB and the CMS-EA code was collected from [119]. The objective function of (4.51) was formed as a function of the force error (difference between desired and actual force). The object function F_t is applied to the CMS-EA based optimisation. The function F_t leads the searching process to achieve the control objective (minimise the force error). The initial population search was selected with $[M_I, B_I, K_I] = [1, 1, 10]$. The other parameters are: number of objective parameters, $N = 2$, initial point of objective variable, $X_{mean} = [0.5, 0.5]$, $\lambda = 4 + 3 \ln(n)$, $\mu = \lambda/4$. Two sets of different initial impedance parameters are used for the CMA-ES based evolution. The initial values of M_I, K_I are only chosen where values of B_I are computed by (4.16) and (4.19). Two different sets are chosen as $M_{Io}, K_{Io} = [0.5, 0.5]$ and $[5, 5]$. The object stiffness values is considered as $k_e = 500 N/m$. The global step size, $\sigma = 0.5$ and the algorithm was iterated 1000 times to obtain the impedance parameters from the simulink model.

4.7.4. Results and discussions

Figure 4.21 and 4.22 present the results of estimation parameters and contact forces respectively for initial impedance parameters, $M_{Io}, K_{Io} = [0.5, 0.5]$. In Figure 4.21, the evolution of M_I and K_I are converged (Top figure). The bottom sub figure presents the fitness function F_t , the best fitness function Best F_t and the standard deviation σ with 160 times iteration. The impedance parameter values are found from the iterations as $M_I, K_I = [0.0050, 4.7561]$. These parameters are applied to the FIM model and the contact forces are achieved for different force reference F_d , shown in Figure 4.22. In all cases, the forces reach the steady state within 0.1 second after the contact. The overshoots were found only $\sim 1\%$ for all three forces.

Figure 4.23 and 4.24 present estimation parameters and contact forces respectively for the initial impedance parameters, $M_{Io}, K_{Io} = [5, 5]$. The values of M_I, K_I are obtained as $M_I, K_I = [0.0011, 3.1387]$. The same force references are used to see the contact force responses when the Barrett Hand hits the object (see Figure 4.24). The all three contact forces reach the final steady state values within 0.1 second after the Barrett fingertip makes contact with the object. The overshoots are found between 2 – 3% of the final values for three desired forces.

The comparison between the force results of Figures 4.22 and 4.24 are presented in the Table 4.2. The table shows the comparison for identical force references with two different values of

4.7 Evolutionary algorithm based impedance control

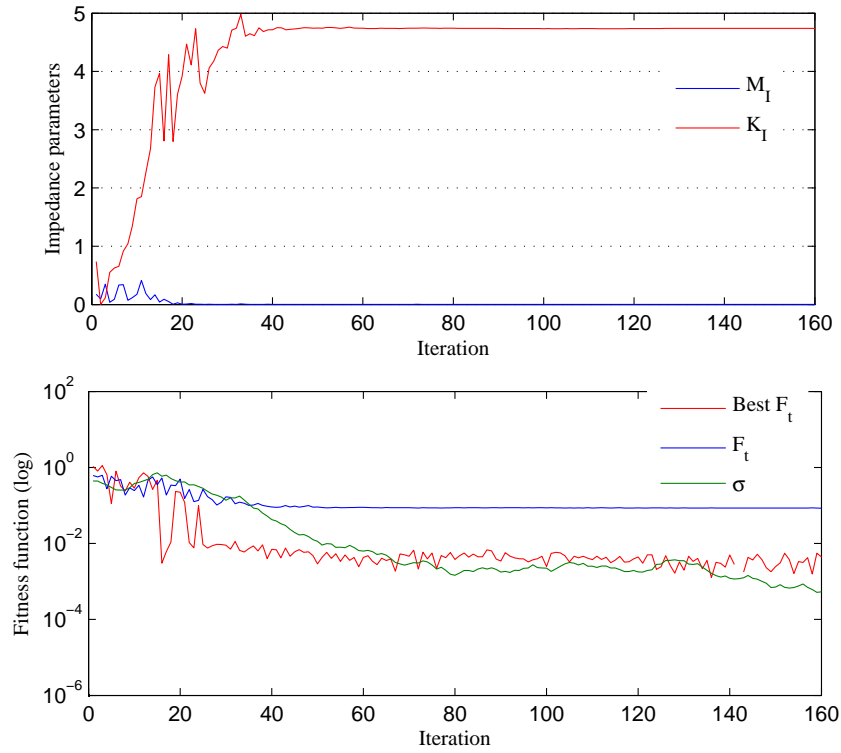


Figure 4.21.: Impedance parameter estimation (Top); Fitness function, standard deviation computation (Bottom) with $M_{Io}, K_{Io} = [0.5, 0.5]$.

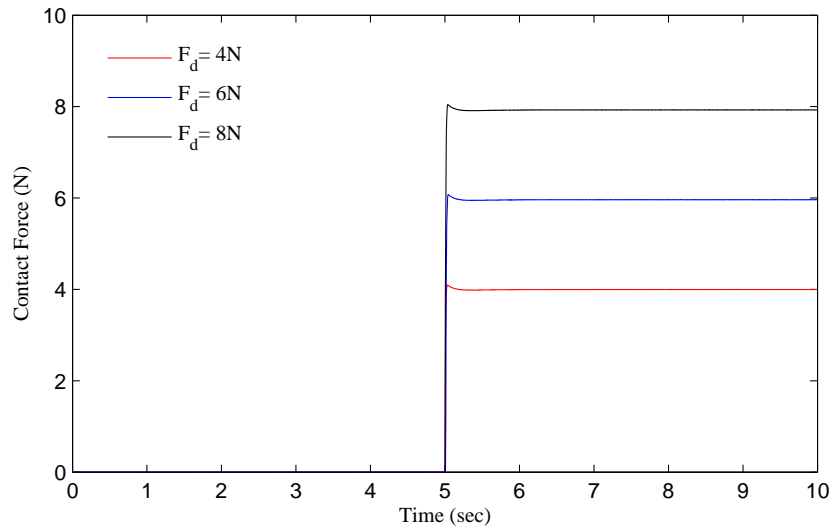


Figure 4.22.: Contact force optimisation with estimated impedance parameters through evolution.

4.7 Evolutionary algorithm based impedance control

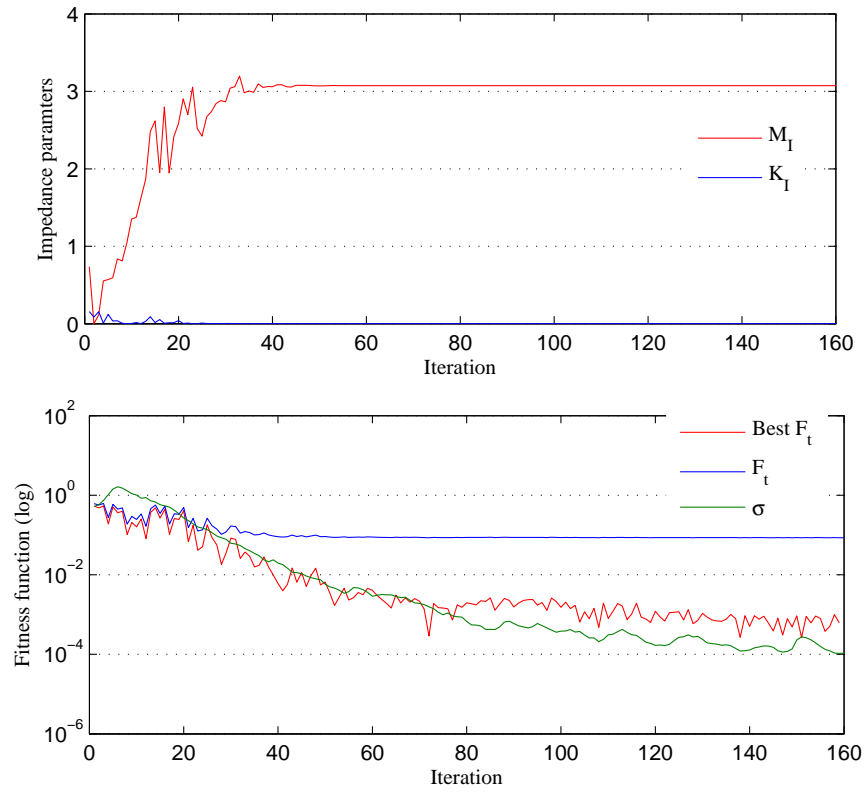


Figure 4.23.: Impedance parameter estimation (Top); Fitness function, standard deviation computation (Bottom) with $M_{Io}, K_{Io} = [5, 5]$.

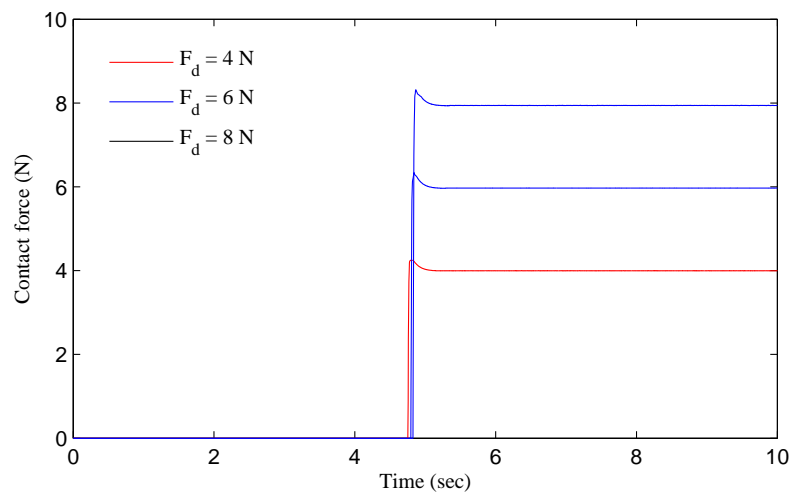


Figure 4.24.: Contact force optimisation with estimated impedance parameters through evolution.

Table 4.2.: Impedance and admittance function for spring and spring damper model.

Desired Force $F_d(N)$	Initial impedance parameters M_{Io}, K_{Io} (kg, N/m)	Final impedance parameters M_{Io}, K_{Io} (kg, N/m)	Settling time (second)	Overshoot (%)
[4, 6, 8]	[0.5, 0.5]	[0.0050, 4.7561]	~ 0.1	~ 1
[4, 6, 8]	[5, 5]	[0.0011, 3.1387]	~ 0.1	2 – 3

M_{Io}, K_{Io} . In both cases, the settling time to reach the desired force F_d is around 0.1 second after the contact. The overshoots are obtained less ($\sim 1\%$) with smaller values of M_{Io}, K_{Io} whereas the overshoots are between 2 – 3% for $M_{Io}, K_{Io} = (5, 5)$. This suggests that, choosing the small initial impedance parameter values reduce the overshoot in achieving the actual contact forces. It is apparent that settling time does not get affected by different initial conditions. This implies that steady state responses are achieved without time delay for different force references but overshoot becomes large with increased values of M_{Io}, K_{Io} .

4.8. Summary

In this chapter, the interactional control problem between the robotic hand and the environment is discussed. A contact model is developed between the Barrett finger and an object. Force control methods are discussed considering the interaction and the advanced features of the impedance control are depicted. The impedance control method is introduced to deal with the contact force during the interaction between hand fingertip and the object. Position and force based impedance controllers are designed and implemented into the Barrett-hand object environment. The simulation results of the contact forces without controllers are obtained and discussed in the section 4.5. The contact force responses are found large with higher object stiffness values and the presence of high damping at the finger joints. In section 4.6, position and force based impedance controllers applied to the planar and the Barrett hand. Contact forces are achieved within a small range (4 – 8N) with PIM method whereas any desired forces are obtained with FIM method. To achieve the force tracking in an unknown environment, adaptation is necessary. Adaptation mechanisms are added in FIM to estimate the unknown environment parameters such as the object stiffness and fingertip location during the contact. The adaptation based results considering an unknown stiffness, also achieved the desired contact forces.

Impedance control parameter selections are not straightforward in any type of impedance based control system. Hence, a CMA-ES based evolutionary algorithm is implemented by an objective function and impedance parameters are evaluated through this optimisation algorithm. Sec-

tion 4.7 explains the process to determine the desired impedance control parameters M_I, B_I, K_I with CMA-ES based optimisation. The parameters are evaluated through CMS-EA and applied to the impedance controller for Barrett hand-object interaction. The results of estimated impedance parameters are presented and discussed in the previous sections.

In this chapter, the concepts of impedance control are brought into the constrained scenario between the robot and the object. The impedance control frameworks are implemented with an objective to minimise the contact forces during interaction. Adaptation is integrated with the existing controllers to deal with unknown environments. Evolutionary methods are considered for impedance parameter selection. Adaptation and CMS-EA based parameter evaluation are the new integration to the impedance controllers. These inclusion would be vital to tackle the contact force problem in complicated grasping and manipulation environment.

5. Classical and adaptive controller implementation for grasping and manipulation

5.1. Introduction to grasping and manipulation

Grasping and manipulation is understood by grasping an object and moving it from one configuration to another with open serial-chain manipulator end-effectors such as the gripper, pincher or multi-fingered robotic hand, then manipulating it according to the desired task planning. The grasping and manipulation is seen as a process to constrain the motion at the object frame with respect to the end-effector frame. The mathematical development of this process requires the knowledge of geometrical relationship of the end-effector-object system, the contact locations, the object, the finger kinematics, the joint link and the fingertip geometries. In this chapter, multi-fingered Barrett hand is used to present the theoretical properties of grasping and manipulation. To discuss the compliant robot hand-object mechanics, it requires understanding of the following:

1. Mathematical model of the object. (kinematics & dynamics)
2. Mathematical model of the robotic hand. (kinematics and dynamics)
3. The contact mechanics between the object and the hand.

The above mathematical properties are used in this study to determine the relationship between motions and forces of the object and the hand. Under the following assumptions,

1. The object is a rigid body and the hand fingers are rigid links.
2. The hand fingers do not slip on the object surface.
3. The contact locations on the object surface are known.

Considering the above assumptions, the grasp kinematics, the grasp Jacobian and the grasp constraint are derived.

5.2. Definition of Grasping

A grasp is defined as a combination of hand posture and position and generates a contact set between the hand and the grasped object. The exact location of these contacts generally depends on the hand position and posture, and also the geometrical shape of the object that are arbitrary and impossible to parameterise. Therefore, the set of contacts cannot be formulated analytically as a function of the variable that defines the grasp though the behaviour of contact sets can be analysed by grasp Jacobian [122]. For computing the grasp jacobian, contact models between hand and object need to be described first. It can be shown as mapping between forces exerted by the fingers at contact point and the resultant wrench at some reference point on the object.

5.2.1. Grasp planning

Grasp planning refers to how the object will be grasped by the hand. In other words, it deals with the problem to identify the position of the finger placement on the object surface. There are different methods available for grasp planning. Planning of grasp is numerically intensive due to the large configuration space. The planning is also subject to the incorporation task-oriented and geometric constraints.

Grasp planning begins with the kinematic analysis of the multi-fingered robotic hand with a given model of an object. The analysis requires a set of robot fingers based on palm, a contact model between the fingers and the object to determine the relationship between motion and the forces of the fingers and the object. Specifically, the interest is to calculate the fundamental grasping constraint which governs the motion of the hand. The assumption made in studying the kinematics is that the contact location between the hand and the object is known, the fingers do not slip on the object, the object and finger links are rigid body, accurate model of the fingers and objects are known. The grasp planning problem is to find suitable contact locations in object and hand finger frame so that a better grasp can be achieved.

5.2.2. Grasp properties

The essential grasping properties include: external force resisting ability whereby the hand fingers should be able to apply contact forces which generate an opposing wrench to applied wrench on an object. The grasp in which hand fingers can resist arbitrary external forces is called the force-closure grasp. The object should be able to move in all directions or some specific directions independently for a desired task. The grasp in which the fingers generate arbitrary object motion called manipulable grasp. To illustrate the grasp properties, consider the basic scenario of an object grasped by the multi-fingered Barrett hand.

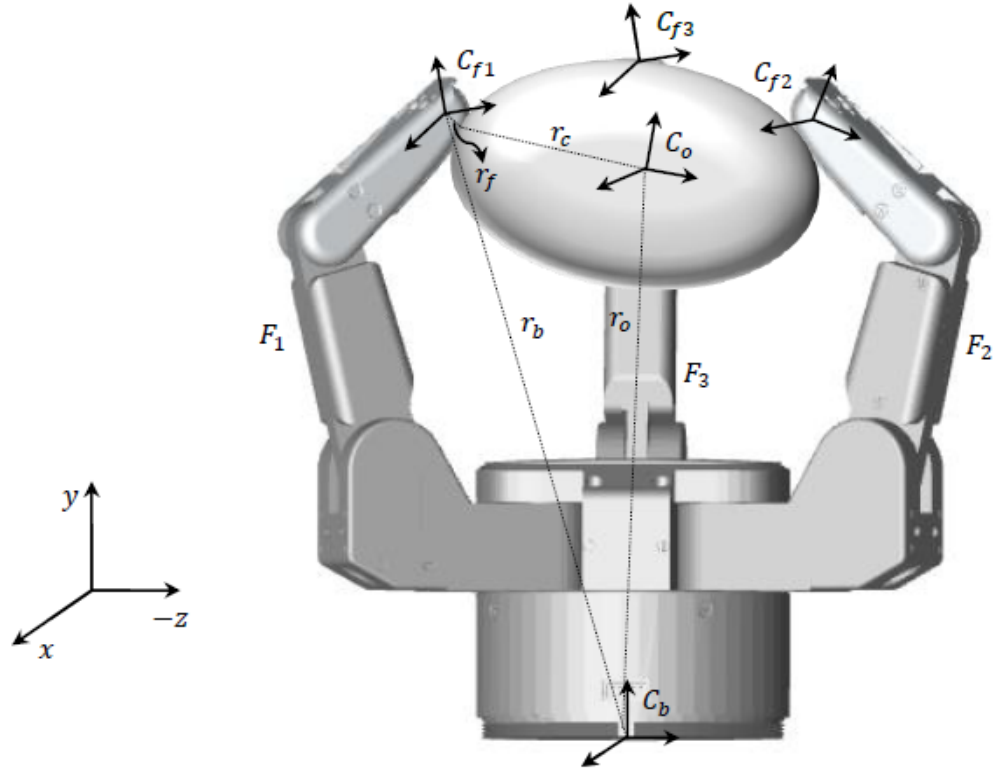


Figure 5.1.: The Barrett hand grasping an object.

Now define the grasp statics based on Figure 5.1. Three contacts between the hand and the object are counted as point contacts at known location on the object's surface. Therefore, the grasp statics expresses the transmission of forces between a set of contacts and the object. A contact between the hand finger and the object is defined as mapping between the exerted fingertip force at the contact point and the resultant wrenches at the center of the mass of the object. Each contact type can be frictionless, frictional or soft contact. The different contact types are shown in Figure 5.2.

A finger exerts force only along the normal of the contact point in a frictionless contact and the applied wrench is represented as,

$$F_i = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} f_i \quad f_i > 0, \quad (5.1)$$

where $f_i \in \mathbb{R}$ is the force magnitude in the normal direction. At each contact, three forces and three moments are stacked by 6×1 vector, which is called the wrench basis vector.

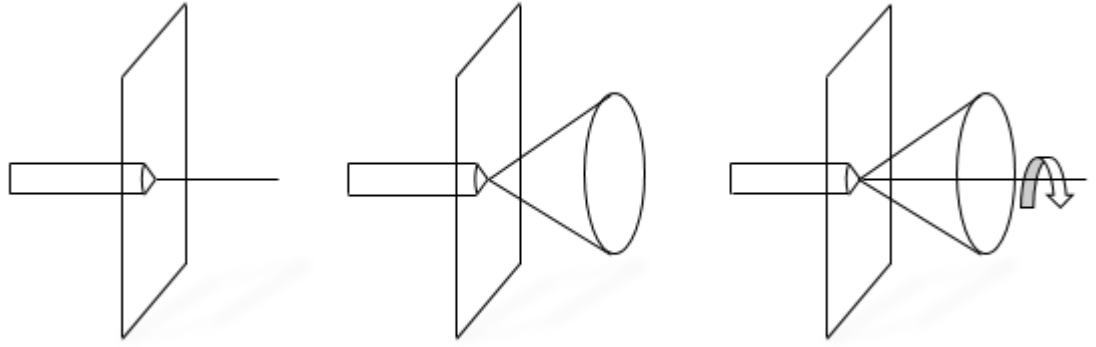


Figure 5.2.: Point contact (Left), Point contact with friction (Middle) and Soft contact (Right).

A frictional model is obtained when friction occurs between the hand fingertips and the object. A frictional contact has both normal and tangential force as shown in Figure 5.3. According to Coulomb's law, the slipping occurs when,

$$f^t > \mu f_n. \quad (5.2)$$

Therefore, the following condition is required for the tangential force to be applied at the contact point,

$$f^t \leq \mu f_n. \quad (5.3)$$

The forces applied at the contact must lie in a cone centered about the surface normal. This is called the friction cone denoted by α (See Figure 5.3) and can be shown as,

$$\alpha = \tan^{-1} \mu. \quad (5.4)$$

In the case of contact with friction, forces are exerted in any direction by staying within the friction cone. The wrench applied to object with the friction model is given as,

$$F_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} f_i \quad f_i \in FC_i, \quad (5.5)$$

where

$$FC = \{f \in \mathbb{R}^3 : \sqrt{f_1^2 + f_2^2} \leq \mu f_3, f_3 \geq 0\}. \quad (5.6)$$

The control algorithms implemented in this chapter are based on point and friction contact.

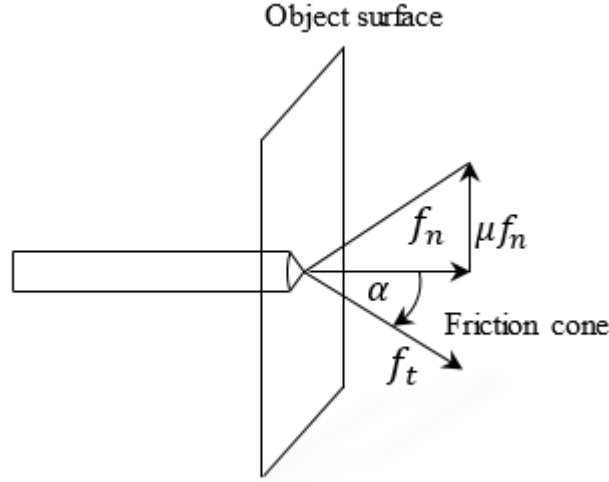


Figure 5.3.: Contact model based on Coulomb's friction.

A soft finger contact also allows the finger to exert a moment about the normal at a point of contact. Each unit wrench corresponds to the normal force f_n , the tangential force f_t and the moment about the normal f_θ ; the magnitude or intensity applied by finger in the normal direction is f_n, f_t, f_θ . This model is not considered for control application. In general, contact wrench applied to the surface of the object can be defined as,

$$F = Wf \quad f \in FC, \quad (5.7)$$

where $W \in \mathbb{R}^{p \times m}$ is the the wrench basis. In $SE(3)$, the force dimension p is chosen as 6. For planar grasping, $p = 2$ indicates the horizontal and vertical direction only. m is the number of forces that is applied through contact. The friction cone FC satisfies the following properties:

1. FC is a closed subset of \mathbb{R}^n .
2. $f_1, f_2 \in FC \implies \alpha f_1 + \beta f_2$, for $\alpha, \beta > 0$.

5.2.3. Grasp map

In the previous section, the contact wrench was determined. The contact wrench is required to transform at the object coordinate frame in order to determine the object wrench. Grasp matrix is responsible for the transformation between the contact and the object's coordinate frame. To derive the grasp map G , set the object coordinate frame as C_o , the center of the mass of the object. C_b is the hand base frame and $C_{f,i}$ is the contact frame where i is the number of contacts. If each finger applies a force $f_{c,i}$ at the contact point $C_{f,i}$, then the applied object wrench is found as,

$$F_o = \sum_{i=1}^n A_i W_i f_{ci} \quad f_{ci} \in FC, \quad (5.8)$$

5.2 Definition of Grasping

where $F_o = [f_o^T \ m_o^T]^T$ is the external force exerted on the object. The terms resultant forces f_o and moments m_o are given as,

$$f_o = \sum_{i=1}^n f_{ci} \quad m_o = \sum_{i=1}^n (r_{oi} \times f_{ci}). \quad (5.9)$$

The term A_i is called the wrench transformation matrix defined as,

$$A_i = \begin{bmatrix} R_{oci} & 0 \\ (r_{oi} \times) R_{oci} & R_{oci} \end{bmatrix}. \quad (5.10)$$

The above equation express the contact wrenches as object wrenches. R_{oci} is the transformation matrix of the contact frame with respect to the object frame. The term $r_o = [p_o, o_o]$ is the position vector from the origin of C_o to the contact point C_c with respect to C_b , where p_o is the position and o_o is the orientation vector. $[r_{oi} \times] \in \mathbb{R}^{3 \times 3}$ is a skew symmetric matrix by the cross product of r_{oi} . The map which transforms the contact forces f_{ci} to the object frame is called the grasp map G which can be expressed as,

$$G_i = A_i W_i, \quad (5.11)$$

where $G \in \mathbb{R}^{6 \times 3}$. Considering grasp map, (5.8) can be rewritten for all finger contacts on the object surface as,

$$F_o = G f_c \quad f_c \in FC. \quad (5.12)$$

The complete grasp matrix is shown as $G = [G_1 \ G_2 \ \dots \ G_n]$ and the contact force set, $f_c = [f_{c1}^T \ f_{c2}^T \ \dots \ f_{cn}^T]^T$. The grasp matrix G is analysed based on the contact point location and the current orientation of the object. In summary, a grasp is defined as a combination of robotic hand fingers exerting forces on an object. The total wrench of the object is calculated from each contact forces generated by the robot finger following the relationship of (5.12). Grasp map becomes valid for contact where rolling or sliding does not occur. The way to ensure no slippage is to satisfy that the contact forces lie within the friction cone at all times.

5.2.4. Force-Closure of a grasp

Another important requirement of grasping is to resist any applied wrench. If a grasp is able to maintain external wrenches by applying accurate robot finger wrenches is called force-closure. The force-closure grasp withstands any applied wrench to the object. If the contact model is defined as point contact with friction, the applied finger forces need to remain in the friction cone in order to achieve the force-closure.

Definition 5.1 Force-closure grasp

5.2 Definition of Grasping

A grasp is called the force-closure grasp for any external wrench $F_e \in R^p$ applied to the object, there always contact forces $f_c \in FC$ occurs such that

$$Gf_c = -F_e. \quad (5.13)$$

Equation (5.13) characterises a grasp is force closure if and only $G(FC) = R^p$. Force-closure properties of the grasp allows the existence of internal forces in grasping and manipulation framework. When an object is grasped by the robot fingers, a set of contact forces occurs in the compliant system which results in no net force on the object or the hand. This set of contact forces is defined as the internal force f_i represented subject to wrench basis W at each point of contact. This force is used in the hand-object system to assure that $f_c \in FC$.

5.2.5. Hand Jacobian

In general, the fingertip forces are expressed in contact coordinate frame on the object surface. The hand Jacobian is required to generate the forces at the fingertip frame. Hand Jacobian is the map which transforms the contact force at the finger frame. Finger kinematics from chapter 2.3 is used to derive the general fingertip Jacobian of the robot fingers with respect to the base frame by the following relation:

$$\dot{X}_i = J_{fi} \dot{\theta}_i, \quad (5.14)$$

where $J_f = \frac{d}{d\theta}(X_i)$; X_i is the fingertip position of each finger in $SE(3)$. Now, it is necessary to transform the wrench from the contact coordinate frame C_c to the base frame C_b . During contact, the resultant wrench for the forces f_f and the moments m_f at each fingertip can be expressed as,

$$F_f = \sum_{i=1}^n W_i^T B_i f_{ci}, \quad (5.15)$$

where

$$B_i = \begin{bmatrix} R_{bci}^T & -R_{bci}^T(r_{ci} \times) \\ 0 & R_{bci}^T \end{bmatrix}, \quad (5.16)$$

is the transformation matrix which expresses the contact coordinate frame C_c subject to the base coordinate frame C_b . From (5.13) and (5.14), the hand Jacobian J_h can be calculated as,

$$J_{hi} = W_i^T B_i J_{fi}. \quad (5.17)$$

In (5.15), $W^T B$ is the transformation of fingertip frame C_f to the contact coordinate frame C_c with respect to the base frame C_b that is multiplied by the fingertip Jacobian. How the grasp map G and the hand Jacobian J transform the different frame in force and velocity space is summarised in Figure 5.4.

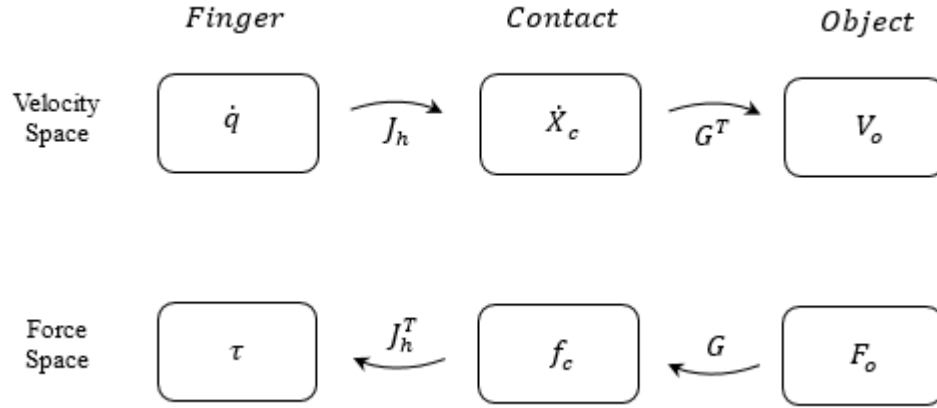


Figure 5.4.: Frame transformation between object and velocities in a multi-fingered grasp.

5.2.6. Grasp constraint

The grasp constraint is essential to derive object grasping which describes the motion of the robot-object compliant system while contact is made between them. This describes the relationship between fingertip and object velocity. As the contact locations are assumed to be fixed contact with friction on the object, the constraint between the hand fingers and the object can be determined by considering the following relation,

$$\dot{r}_{oi} = \dot{r}_{ci}, \quad (5.18)$$

where \dot{r}_{oi} is the object and \dot{r}_{ci} is the fingertip velocity at the i -th contact point. Applying (5.11) and (5.16) to (5.17) with respect to the base frame, the grasp constraint holds the following relation,

$$G_i^T \dot{x}_o = J_{hi} \dot{q}_i, \quad (5.19)$$

which is the relationship between the object velocity and the angular velocity of the i -th finger. This constraint regulates the grasping and manipulation of an object. The velocity vectors of the object is, $\dot{x}_o = \begin{bmatrix} \dot{r}_o^T & \omega_o^T \end{bmatrix}^T \in \mathbb{R}^6$, where ω_o is the angular velocity of the object with respect C_b . The spatial Jacobian has the form, $J_{hi} \in \mathbb{R}^{6 \times l} = W_i^T B_i J_{fi}$ where l_i is the link number of the fingers and J_i is the kinematics Jacobian at the i -th fingertip.

5.3. Robot Object dynamics

Grasp constraint is determined in the previous section which is required to derive the dynamics of robotic hand grasping an object manipulating an object. From (5.6), the dynamics of each

finger of the robotic hand have the following form,

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau - \tau_d, \quad (5.20)$$

where τ_d is defined as $J^T f_c$ which is the reaction torque due to the force $-f_c$ that the object is exerted on the hand fingertip. Considering the jacobian transpose, (5.20) is rewritten as,

$$\ddot{q} = M_f(q)^{-1} [\tau - C_f(q, \dot{q})\dot{q} - G_f(q)] \quad (5.21)$$

and the object dynamics from equation (2.15) is given as,

$$M_o\ddot{X} + C_o(X, \dot{X})\dot{X} = F_o = Gf_c + f_e. \quad (5.22)$$

Now rearrange the constraint equation (5.19) by dropping subscripts as,

$$\dot{q} = J_h^{-1} G^T \dot{x}_o. \quad (5.23)$$

Differentiating the equation (5.23) yields,

$$\ddot{q} = J_h^{-1} G^T \ddot{x}_o + \frac{d}{dt} (J_h^{-1} G^T) \dot{x}_o. \quad (5.24)$$

Using (5.23), the robot-object dynamics are formed from (5.21) and (5.22) as,

$$\tilde{M}(q, x)\ddot{x}_o + \tilde{C}(X_o, \dot{X}_o)\dot{x}_o + \tilde{N}(X_o, \dot{X}_o) = F, \quad (5.25)$$

where

$$\begin{aligned} \tilde{M} &= M_o + GJ_h^{-T} M_f J_h^{-1} G^T \\ \tilde{C} &= C_o + GJ_h^{-T} (C_f J_h^{-1} G^T + M_f \frac{d}{dt} (J_h^{-1} G^T)) \\ \tilde{N} &= N_o + GJ_h^{-T} N_f \\ F &= GJ_h^{-T} \tau. \end{aligned}$$

Equation (5.25) is determined assuming that the joint motions q, \dot{q} are derivable from the object motions X, \dot{X} . The terms \tilde{M} is the effective mass matrix of the object, \tilde{C} is the effective Coriolis matrix and \tilde{N} as the effective gravitational forces. The hand dynamics and the object dynamics combines the effective compliant matrix terms.

Definition 5.1 Dynamic properties of the robot-object equations of motion

1. $\tilde{M}(q, x)$ is symmetric and positive definite.
2. $\ddot{\tilde{M}}(q, x) - 2\tilde{C}$ is a skew – symmetric matrix.

The internal forces explained in the section 5.2.4 exist in robot-dynamic equations of motion such as equation (5.24). The term $F = GJ_h^{-T} \tau$ includes the presence of internal forces belong-

ing to the null space of grasp matrix G . This term transforms the finger joint toques into object forces.

5.3.1. Multi-fingered Grasping and manipulation problem formulation

The assumptions for controller designs are as follows: each manipulator is non redundant, that is, the number of degree of freedom of each manipulator is equal to the dimension of operational coordinate. The hand fingertips and the object are required to be rigid to have fixed point contact with friction during interaction. The kinematics and dynamics of the links and the object are known. All end-effectors of the manipulators are rigidly attached to the common object so that there is no relative motion between the object and any end-effector. During the coordinated motion, the manipulators do not enter any singular configuration. The grasping and manipulation control problem is stated as: the control of the motion (position and orientation) of the object in \mathbb{R}^3 by the joint torque τ_i of the fingers of the hand. When fingertips make contact with the object, the joint torque τ_i is transformed to the fingertip forces F_f which causes the object to move according to the command.

- **Hand workspace tracking:** Given a desired contact location trajectory $X_{DC} \in R^{6 \times 1}$, each fingertip of the robotic hand X_C should track X_{DC} . The first objective is,

$$\lim_{t \rightarrow \infty} (X_{DC} - X_C) = 0. \quad (5.26)$$

- **Object Tracking:** Given a desired object trajectory $X_{DO} \in \mathbb{R}^6$, the centre of mass of the object $X_O \in \mathbb{R}^6$ should track X_{DO} while the contact is made between the hand and the object considering the grasp constraint. The second objective is then,

$$\lim_{t \rightarrow \infty} (X_{DO} - X_O) = 0. \quad (5.27)$$

- **Contact Force Tracking:** The actual contact force $F_C \in R^9$ should follow the desired contact force $F_D \in R^9$ and lie within the friction cone at all times. The third objective is,

$$\lim_{t \rightarrow \infty} (F_D - F_C) = 0. \quad (5.28)$$

Robot fingers are tracked in the Cartesian space for desired contact location before the object is grasped. Object tracking problem occurs when the hand fingertips reach the contact locations, then the contact force is generated which needs to be optimised.

Maintaining contact without slipping: The contact should be maintained over the entire time of the trajectory by producing non-zero contact forces which satisfies the friction cone properties. It can be also presented that an internal force f_i is found for an arbitrary set of

contact force f_c such that $f_c + f_i$ lies in the friction cone. The force f_i have no net effect on the motion of the hand or object and is available in the null space of the Grasp matrix, G .

5.4. Computed torque controller

The control purpose of the multi-fingered hand is to grasp an object and ensure damages are not caused to the object. Implementing a controller for a robot-object compliant system is challenging as it is a multiple-input-multiple-output (MIMO) nonlinear system with many actuators and sensors. For control analysis, consider the constrained robot-object dynamics of (5.25) as,

$$\tilde{M}(q, x)\ddot{X}_o + \tilde{C}(X_o, \dot{X}_o)\dot{X}_o + \tilde{N}(X_o, \dot{X}_o) = F = GJ_h^{-T} \tau. \quad (5.29)$$

The tracking problem is to calculate the joint torque such that the system asymptotically track the desired object manipulation trajectory X_{do} . $GJ_h^{-T} : \mathbb{R}^n \rightarrow \mathbb{R}^p$ is a surjective map which allows for the determination of a set of torques τ . From **Definition 5.1**, the effective mass matrix \tilde{M} and the coriolis matrix \tilde{C} show the same properties that apply to open chain manipulators. Therefore, any type of controllers used for multi-fingered Barrett hand in the Chapters 3 and 4 can be used for grasping and manipulation.

The classical CTC algorithm used in chapter 3.4 is used in this section. The advantages of the CTC is that it linearises a nonlinear dynamical system by using available linear control theories. The CTC is also known as feedback linearisation method where full-state nonlinear feedback is applied to a nonlinear system. CTC has limitations in terms of the input magnitude and the computational time required to use the feedback law to globally convert a nonlinear system into a linear one. In robotic system such as the multi-fingered hand, the inertia matrix is bounded followed by the dynamic properties. This suggests that the input boundedness is not an issue to apply CTC law. Also, the control torque exerted on the constrained system remains bounded.

In addition, experimental results presented in chapter 3 indicate that the CTC is a standard method to implement tracking controller for a robotic hand. When the initial conditions of the object are known and $X_o(0) = X_{do}(0)$, $\dot{X}_o(0) = \dot{X}_{do}(0)$, then the controller can be chosen as,

$$F_o = \tilde{M}(q, x)(\ddot{X}_{do} - K_v \dot{e}_o - K_p e_o) + \tilde{C}(X_o, \dot{X}_o)\dot{X}_o + \tilde{N}(X_o, \dot{X}_o), \quad (5.30)$$

where $e_o := X - X_d$. The proof of asymptotic convergences require that $\tilde{M}(q, x)$ is positive definite and $\tilde{M} - 2C$ is skew symmetric.

Using the jacobian transpose relationship between the fingertip forces and joint torques, force F is determined as $GJ^{-T} \tau$. The general solution of $F_o = GJ^{-T} \tau$ has the following form,

$$\tau_o = J^T G^+ F + J^T f_i, \quad (5.31)$$

where $G^+ = G^T(GG^T)^{-1}$ is the pseudo-inverse of G and $f_i \in \mathfrak{N}(G)$ i.e. the internal forces f_i belongs to the null space of G . As $Gf_i = 0$, f_i can be chosen arbitrarily satisfying the friction properties that does effect the tracking feature of the controller. Considering the solution of (5.31), the CTC law in terms of the joint torque input can be implemented as,

$$\begin{aligned} \tau_o = J^T G^+ & \left[\tilde{M}(q, x)(\ddot{X}_{do} - K_v \dot{e}_o - K_p e_o) \right. \\ & \left. + \tilde{C}(X_o, \dot{X}_o)\dot{X}_o + \tilde{N}(X_o, \dot{X}_o) \right] + J^T F_n. \end{aligned} \quad (5.32)$$

The above law provides the torque required to drive the constrained system along its desired path. After canceling the nonlinearities of the constrained system, the error dynamics in the object frame becomes,

$$\ddot{e}_o + K_v \dot{e}_o + K_p e_o = 0. \quad (5.33)$$

Equation (5.33) is a second order linear equation where the selection of control gains K_p and K_v are easy to choose and can be proved that the system is stable and motion error $e_o \rightarrow 0$ exponentially as time $t \rightarrow \infty$.

5.4.1. Stability of the CTC law

When the selections of control gain $K_p, K_v \in \mathbb{R}^{n \times n}$ are positive definite and symmetric matrices, then the manipulation control law of (5.32) applied to (5.29) results in exponential tracking of object motion X_o .

Proof: The error dynamics of (5.33) can be expressed in matrix form as,

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} &= E \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \\ E &= \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix}. \end{aligned} \quad (5.34)$$

In (5.34), each eigenvalue of the diagonal matrix E is negative real. Let λ be the eigenvalue of E with associated eigenvector $v = v_1, v_2$, then (5.34) becomes,

$$\lambda \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}. \quad (5.35)$$

It is seen from (5.36) that $v = 0$ when $\lambda = 0$ which implies $\lambda = 0$ is not an eigenvalue of E . When $\lambda \neq 0$, then $v_2 = 0$ which means that $v_1 = 0$. Therefore, $v_1, v_2 \neq 0$ and it assumed as

$\|v_1\| = 1$ without the loss of generality. From this assumption,

$$\begin{aligned}\lambda^2 &= v_1^* \lambda^2 v_1 = v_1^* \lambda^2 v_2 \\ &= v_1^* (-K_p v_1 - K_v v_2) \\ &= -v_1^* K_p v_1 - \lambda v_1^* K_v v_1,\end{aligned}$$

where, \star represents complex conjugate transpose. Consider the terms, $\alpha = v_1^* K_p v_1$ and $\beta = v_1^* K_v v_1$ yields,

$$\lambda^2 + \alpha\lambda + \beta = 0 \quad \alpha, \beta > 0,$$

which proves that the real part of the eigenvalue λ is negative.

5.4.2. Computed torque control for Barrett hand grasping and manipulation

The CTC based control algorithm for grasping and manipulation of the Barrett Hand is built in SimMechanics. The full control structure is given in Figure 5.5.

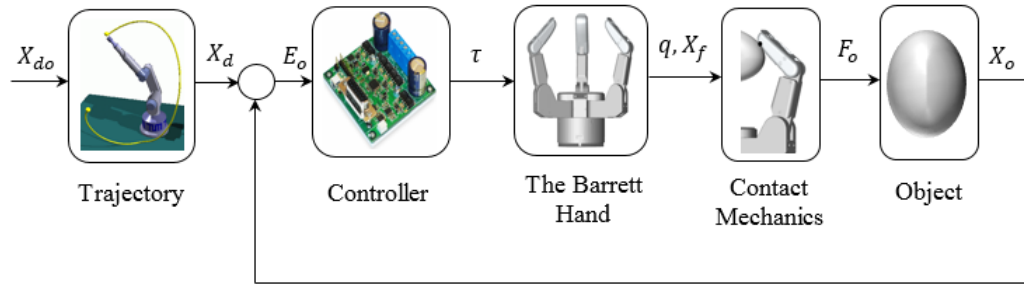


Figure 5.5.: General control structure for grasping and Manipulation by the Barrett hand.

One controller is first applied for workspace robot dynamics before the contact. When the contact is made, another controller is activated to apply to the object space. It is assumed that the object is within the range of the Cartesian space of the Barrett hand. The hand is commanded from its initial fingertip position x_f, y_f, z_f to reach the known contact locations x_{ci}, y_{ci}, z_{ci} ($i = 1, 2, 3$). The simulation system diagram developed for controlling the Barrett hand before contact is shown in Figure 5.6.

The desired step input is supplied through demand input block and the time-based Cartesian fingertip trajectory input X_d is generated. Joint trajectory input q_d is calculated from the inverse kinematics developed in chapter 2.3.2. Feedback control block output is the solution of (5.33) where e_q is the joint error. The term 'Aux Input' is designed after the cancellation of non-linear robot dynamics and linearisation. This drives the feedforward block which is governed by (5.32). This part provides the joint torque τ required to generate the motion to the robotic hand. The robotic hand output q is fed back to the feedback control block and determines the

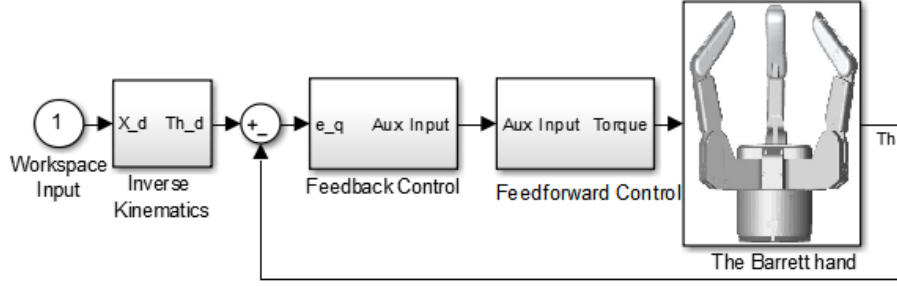


Figure 5.6.: General control structure in the hand coordinates before grasping by the Barrett hand.

joint error e_q . The control structure after grasping is demonstrated in the flow diagram of Figure 5.7. Before the grasp is made, the control layout of Figure 5.6 applies to the system. When all fingertips have reached the desired location on the object's surface, the contact force f_c is produced and the input becomes the desired object motion trajectory X_{do} . The computed torque law of (5.32) is then applied to the Barrett Hand to generate torque and grasp and manipulate the object. The object space motions are output which is fed back and transformed to the joint spaces. The joint spaces control law is then worked as an internal controller of the system.

The simulink structure of the robot-object system before and after grasping is displayed in Figure 5.8. A *switch* block is used as conditions to pass the torques calculated before and after grasping. After grasping the torque τ_o is generated through the contact mechanics and the control law is calculated in the object space. The contact is modelled such that, when $f_c < 0$, controller for the object frame is not activated. The joint space based torque τ is calculated for applying to the robotic hand until all fingertips reach the desired contact location and introduce the contact force.

5.4.3. CTC results and discussion

For the Barrett hand simulation, the mass, the link and other parameters are: link lengths, $l_1 = 0.05m$, $l_2 = 0.07m$, $l_3 = 0.07m$, link mass, $m_1 = 0.03kg$, $m_2 = 0.05kg$, $m_3 = 0.03kg$, base height of the hand, $h = 0.06m$ and gravity, $g = 9.81m/s^2$. Initial and contact fingertip positions of all three fingers are given in Table 5.1.

Table 5.1.: Fingertip positions of the Barrett hand at initial and contact state.

Finger F_i	Initial fingertip position $F_{fi}(m)$	Contact fingertip position $F_{ci}(m)$
F_1	(0.00049, -0.1568, 0.1122)	(0.0255, 0.1712, -0.0292)
F_2	(0.00049, -0.1568, 0.1122)	(-0.0255, 0.1712, -0.0292)
F_3	(0.00049, 0.1568, 0.1122)	(0.00049, 0.1712, 0.0276)

The CTC method is tested in two stages for grasping and manipulation of the Barrett Hand, the

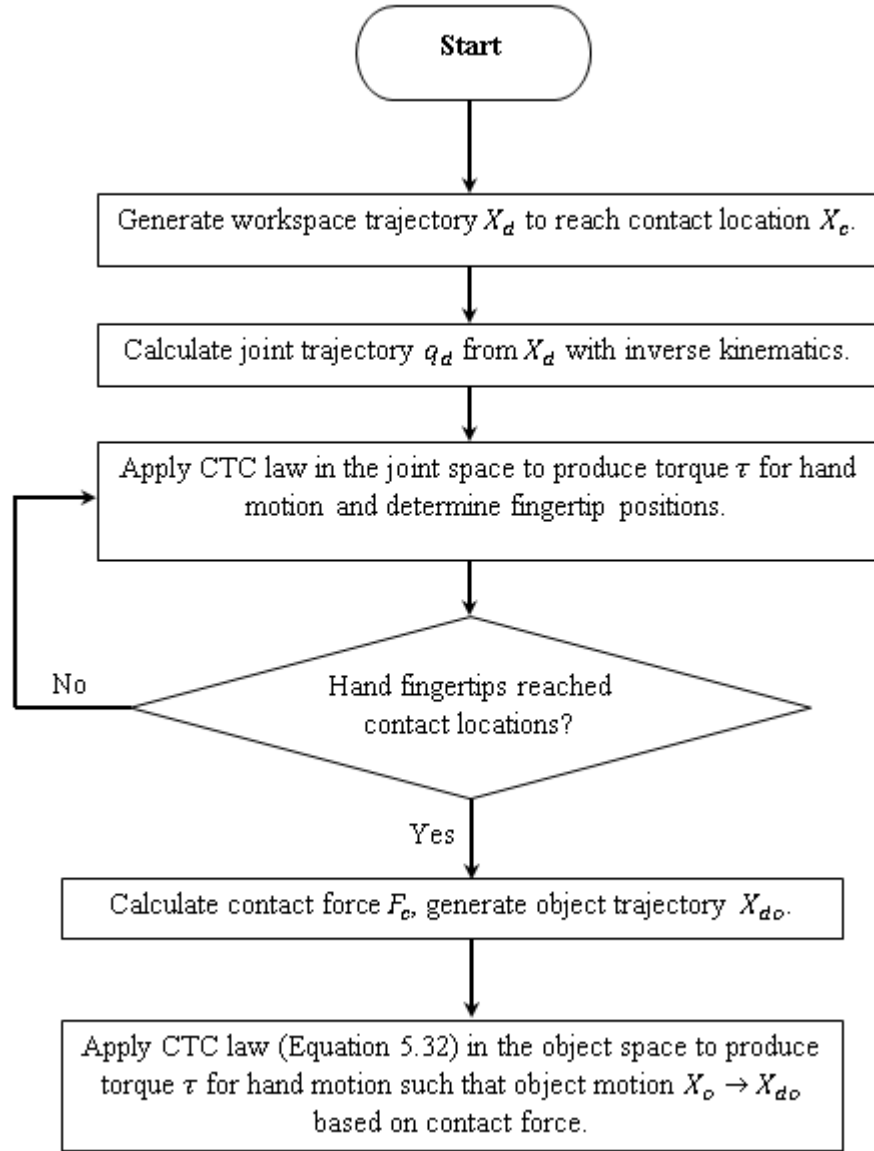


Figure 5.7.: General control structure in the object space after grasping by the Barrett hand.

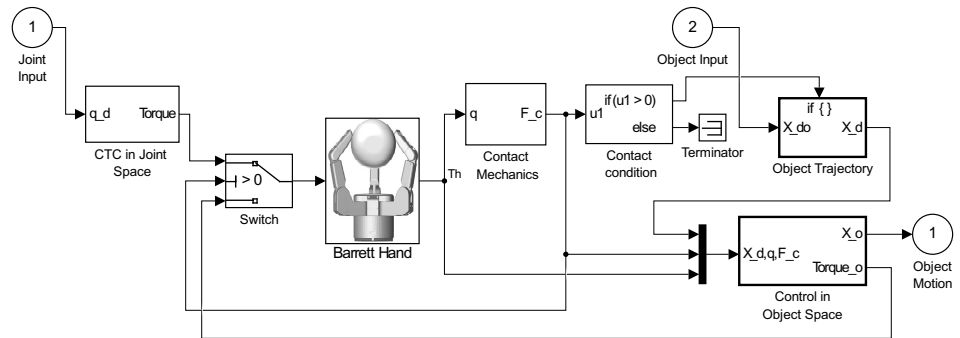


Figure 5.8.: Control structure in the object space after grasping by the Barrett hand.

pre-grasp state and the Grasp state.

I. Pre Grasp State

In pre grasp state or the Cartesian open space, the control gains for joint based CTC are obtained through trial and error. The proportional and the derivative gain are $K_p = 300$ and $K_v = 29$ respectively. For the desired contact locations given in Table 5.1, joint angles q_i are calculated for F_f through inverse kinematics and the joint based CTC law is applied to track the joint trajectory q_{di} . Trajectories are commanded to reach the contact areas on the object at the same time. The trajectory reference is calculated as follows,

$$q_{di}(t) = q_{doi}(1 - \exp^{-t/a}) \quad (5.36)$$

where, q_{doi} is the step input with the final joint reference. Equation (5.36) provide the smooth step input trajectory over time. Spread angle sp_i is valid for F_1 and F_2 only. First, F_1 and F_2 reach at the co-ordinate locations $(0.0255, -0.0292)$ and $(-0.0255, -0.0292)$ respectively. Three fingers then move together to grasp the object at the contact location F_{ci} . Joint motion tracking of three fingers of the Barrett Hand are shown in Figure 5.9. The joint references for all fingers are, $q_2 = 75^\circ$, $q_3 = 35^\circ$, $sp_{1,2} = 1^\circ$. The object grasped at 6.85 seconds. The tracking errors ($\sim 0.5 - 1\%$) are visible for sp_1 and sp_2 before settle down. The other two joints q_2 and q_3 starts moving when sp_1 and sp_2 reach the desired value, $sp_d = 1^\circ$. Joint q_2 for all fingers take less than 1 second to settle down to the desired values ($q_{2d} = 75^\circ$). The joint errors are obtained as $\sim 2 - 3\%$ of the desired values. The joint q_3 of all fingers have a little spike downwards before moving to the reference values ($q_{3d} = 35^\circ$). The spikes in the responses appear due to the presence of damping in q_3 . Despite the presence of damping, the settling time for q_3 responses are also found between 1 – 1.5 seconds. The tracking errors are measured between $\sim 2 - 4\%$ of the references. The largest tracking errors are found from q_3 joints of the finger F_3 .

In Figure 5.10, joint velocity responses of the fingers are shown. The velocities of the fingers are moving towards a steady value and implies that system is dissipating energy and achieving the equilibrium. The torques τ_i are calculated using the CTC law for all fingers and presented in Figure 5.11. In all figures, joint torques became smooth very quickly. Velocity and torque profiles of the hand verify the system model and proves the ability of the fingers to track any desired motions.

The contact mechanics described in chapter 4.2 is used to calculate the contact forces. Normal contact forces F_{ni} obtained from object grasping by the Barrett hand are presented in the Figure 5.12. The object stiffness value is considered, $K_s = 10kN/m$. In all figures, the normal forces are zero before 7 seconds which means that the fingers are still in the open space. As soon as the contacts are made, the contact forces are measured and the grasping mode is activated. Following the coordinate declaration in the Figure 5.1, the *sign* of the contact force values

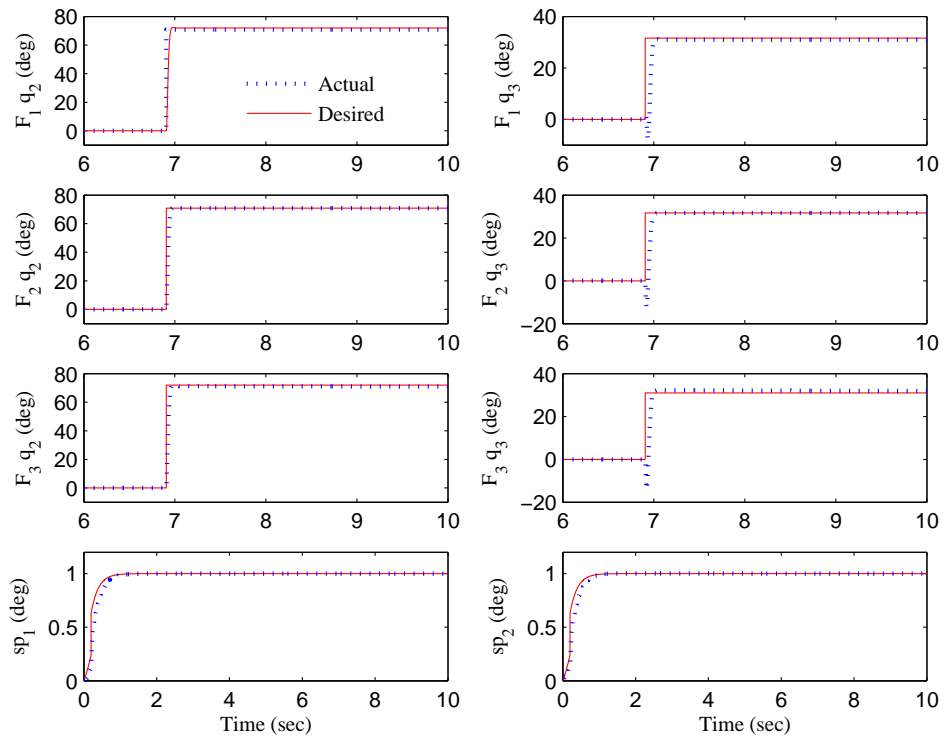


Figure 5.9.: Tracking results of joint motions q_i in the Cartesian space before grasp state.

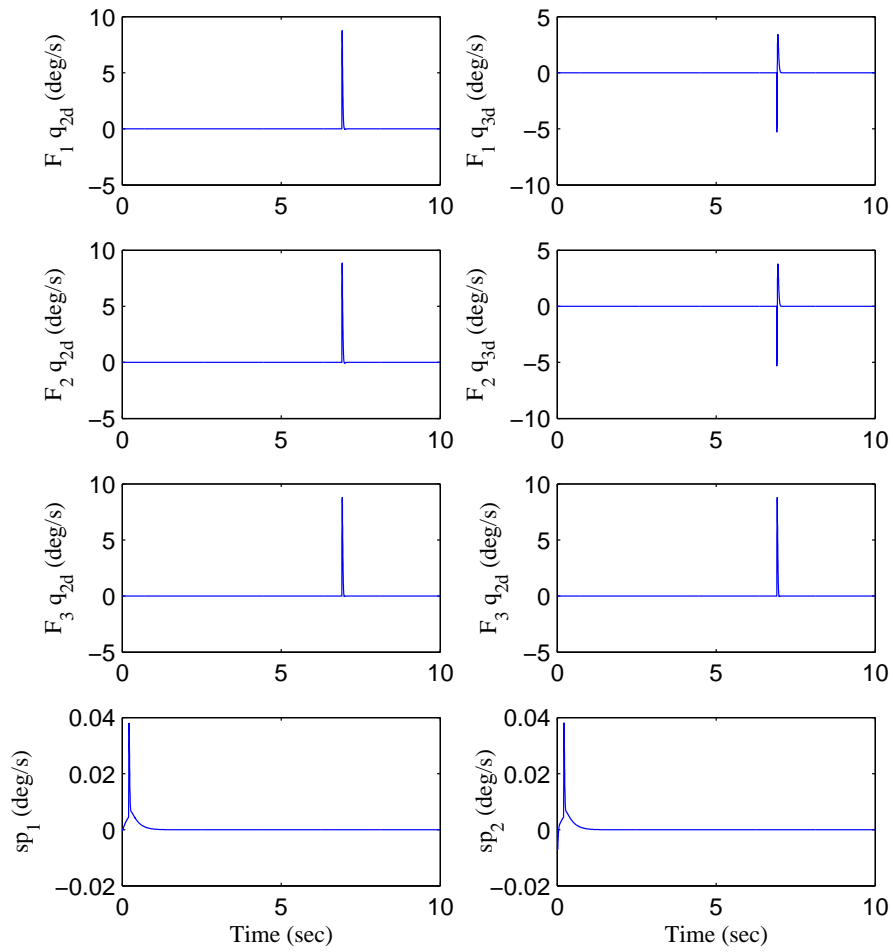


Figure 5.10.: Joint velocities \dot{q}_i of a Barrett finger in the Cartesian space before the grasp state.

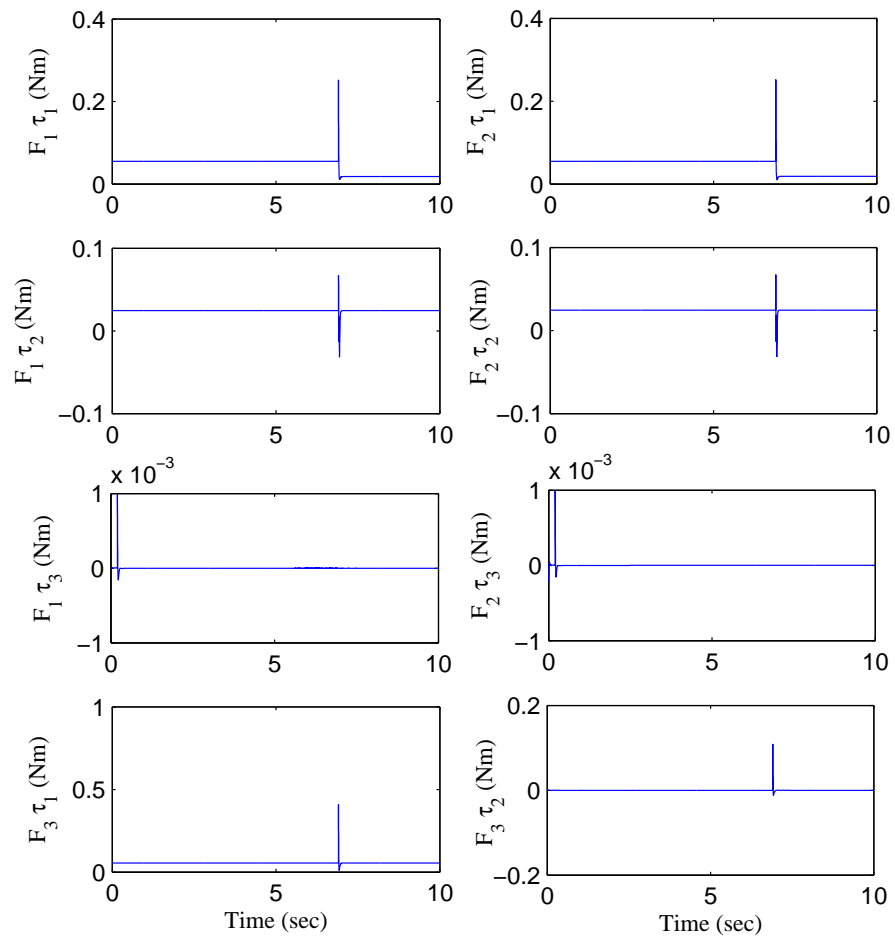


Figure 5.11.: Joint torques τ_i of the Barrett fingers before grasping.

for three fingers are, $[F_1, F_2, F_3] = [-, -, +]$. It is important to have two different signs in the normal forces in order to achieve grasp due to the internal force.

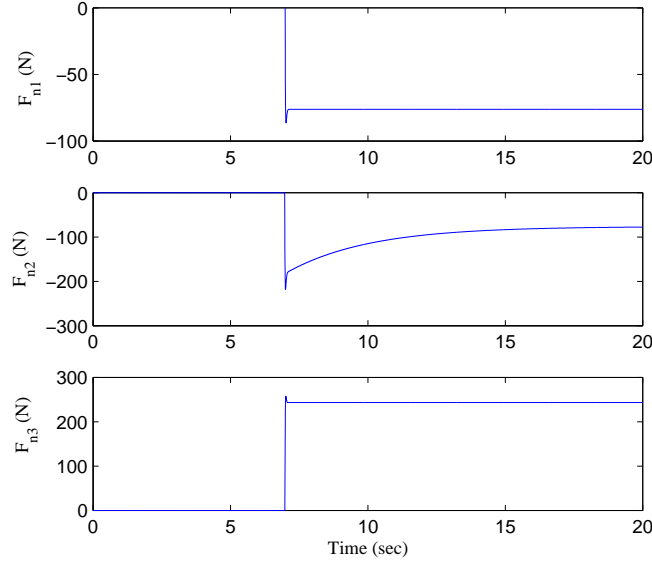


Figure 5.12.: Normal contact forces F_n of Barrett fingers during grasp .

II. Grasp State

The CTC is also for manipulation of the object once it is grasped. The initial and desired object positions are given in Table 5.2. The desired object positions are given subject to the initial object frame C_o . The object mass inertia values are: $I_{xx} = 0.005$, $I_{yy} = 0.0061$, $I_{zz} = 0.0061$. Grasp mode is activated when contact force is measured in the coupled robot-object system. The robot-object compliant subsystem receives the joint data q_i , the grasp matrix G and the Jacobian J .

Table 5.2.: Initial $X_o(o)$ and desired $X_{do}(t)$ positions of the Object for grasping.

Object Position (m)	x_o	y_o	z_o	ϕ_o	θ_o	ψ_o
Initial $X_o(o)$	0.00049	0.1708	0	0	0	0
Desired $X_{do}(t)$	0.0006	0.171	0.01	$2 * \pi / 180$	$\tau / 20$	$\tau / 20$

The controller derived in (5.32) is applied for the simulation of the object manipulation. The results in Figure 5.13 show the performances of motion where desired and actual object motion of the system are compared. The horizontal object position z_o of the object has 0.5 – 1% overshoot when the contact is made and settling down time is found 1.2 seconds. The vertical object position y_o have 1.17% overshoot. No controller applied on the side position x_o and remain constant. Orientation variable ϕ approach to the desired value 0.035 rad . The tracking

errors are shown in the Figure 5.14. The maximum errors for the variables, $z_o = 0.05$ and $y_o = .002m$. The variables ϕ_o and ψ_o are found with maximum errors of 0.05 and 0.005 respectively. The torque input signals τ_{od} applied to the finger joints after grasping is presented in Figure 5.15. The settling times of the torque signals are maximum of 1 sec after the initial oscillation during the contact between the fingertips and the object. No large oscillations in the signals are visible in the complete time history of the simulations.

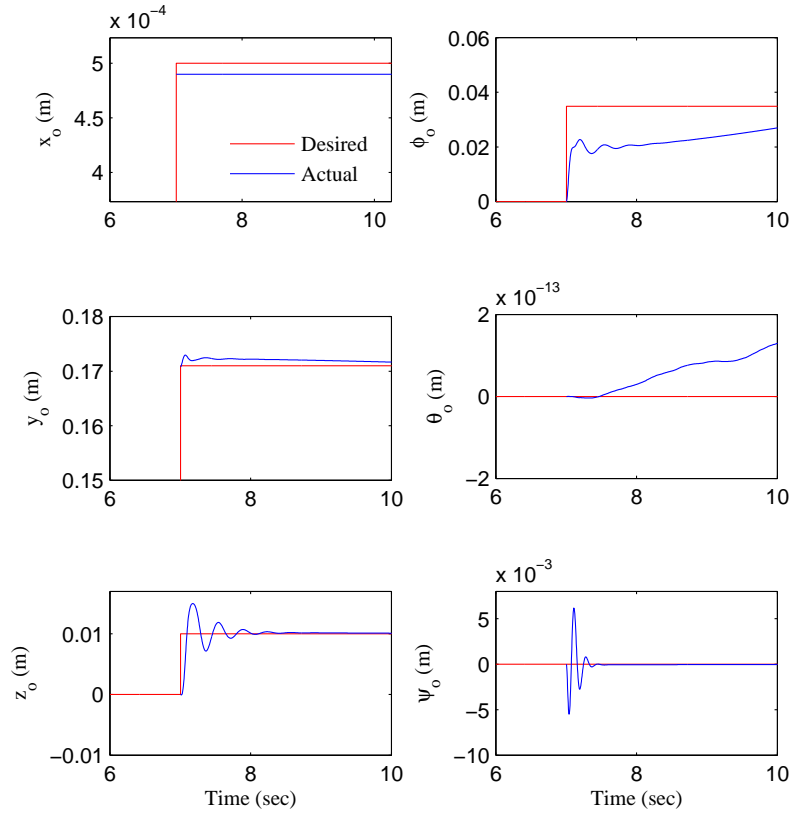


Figure 5.13.: Object motion X_o tracking after grasped by the Barrett hand.

The results presented in Figure 5.9 – 5.14 is summarised in the Table 5.3. The table provides the range between the minimum and maximum values of the overshoot, tracking error and settling time for the joint q_i the object position X_o . The joint motion overshoots stay within 0.1% of the final values. The maximum tracking errors are occurred 4% between the reference and the actual position. Settling time is also found between 1 – 1.5 seconds. Tracking errors are found large in the case of q_3 due to the damping. The overshoot found from the object response is more compared to the joints but have less tracking errors. In both cases, over shoots are less, tracking errors and settling time values are also within a small range by the CTC controller.

- Object motions are well tracked based on contact mechanics and the grasp constraint

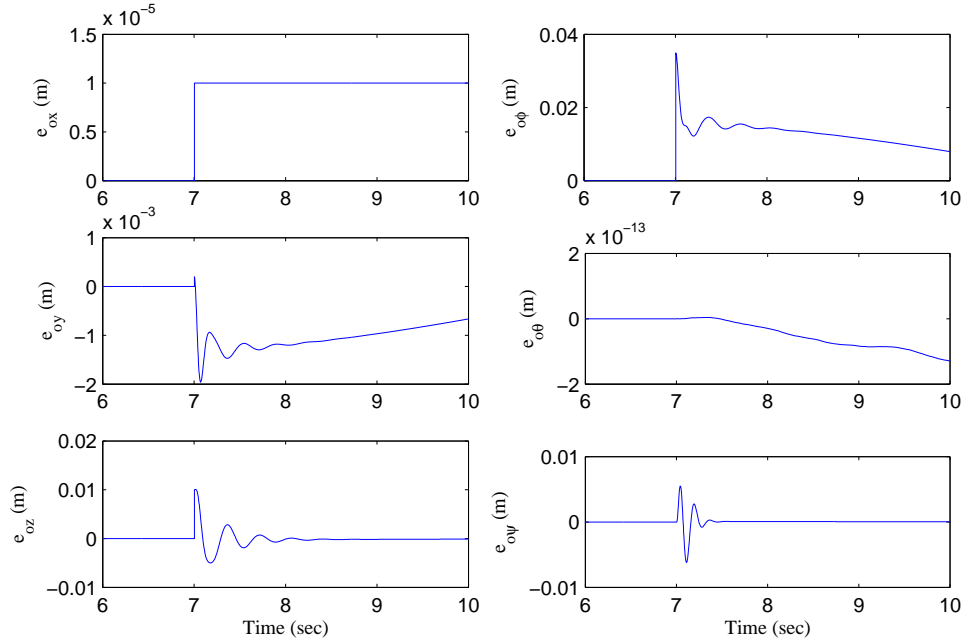


Figure 5.14.: CTC based Tracking error e_o of the object after grasped by the Barrett Hand.

Table 5.3.: Range of settling time, tracking error and torque profiles of the joints and the object (before and after grasp).

Output type	Overshoot (%)	Tracking error (m)	Settling time (second)
Joint motion	$\sim 0.1\%$	0.5 – 4	1 – 1.5
Object motion	0.5 – 1.1%	0.002 – 0.05	1 – 3

once grasp is made. Object actual positions x_o, y_o, z_o are found with little errors. Orientation angles ϕ_o, θ_o, ψ_o of the motions tends to zero but improvements are possible considering other issues such as frictions and internal force control.

5.5. Natural and stiffness controller

Natural control law for compliant based grasping was first used by Koditschek [?]. The algorithm presented in that study is similar to computed torque method but does not follow the way to cancel the non-linear dynamics of the system precisely. The skew symmetric property of the robot dynamics is $p^T(\dot{M} - 2C)p = 0$ for all $p \in \mathbb{R}^n$. This is a modified version of PD control law which can be proved asymptotically stable for motion tracking problem. The natural control algorithm is implemented as,

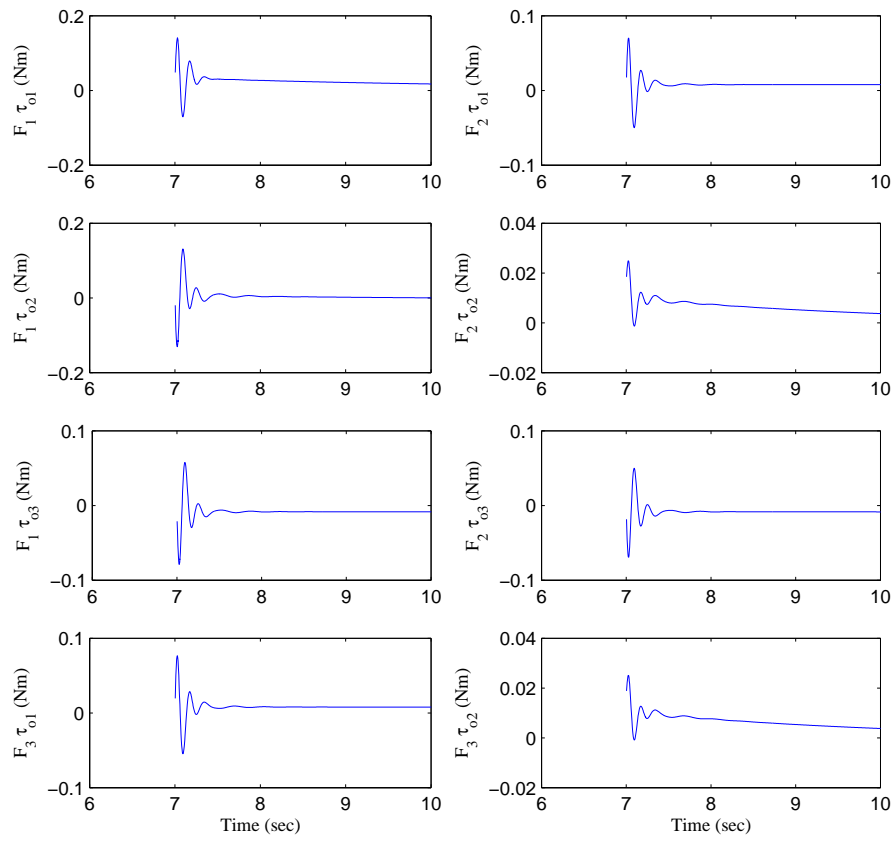


Figure 5.15.: Joint torques τ_i of the Barrett fingers after grasping.

$$\tau_{on} = J^T G^+ \left[\tilde{M}(q, x) \ddot{X}_{do} + \tilde{C}(X_o, \dot{X}_o) \dot{X}_{do} + \tilde{N}(X_o, \dot{X}_o) + \tilde{C}(X_o, \dot{X}_o) \dot{X}_o - K_v \dot{e}_o - K_p e_o \right] + J^T F_n. \quad (5.37)$$

In CTC law of (5.30), the error dynamics is multiplied by $\tilde{M}(q, x)$. But the natural law of (5.36), the motion error and velocity error are calculated separately with control gain. Also, the nonlinear Coriolis matrix $\tilde{C}(X_o, \dot{X}_o)$ is associated with desired object velocity \dot{X}_{do} . The only difference in the stiffness control law is the ability to show exponentially stable error dynamics. The stiffness based grasp control law is given as,

$$\tau_{os} = J^T G^+ \left[\tilde{M}(q, x) (\ddot{X}_o + \lambda \dot{e}_o) + \tilde{C}(X_o, \dot{X}_o) \dot{X}_{do} + \tilde{N}(X_o, \dot{X}_o) - K_v \dot{e}_o - K_p e_o \right] + J^T F_n. \quad (5.38)$$

To achieve the exponential stability over the motion errors, e_o and \dot{e}_o are multiplied by stiffness gain $\lambda > 0$ which is positive definite. The object motion tracking based on the natural controller are presented in 5.16.

5.5.1. Results and discussions

The mass and inertia parameters are used from the previous section for the natural controller. Controller expression is different but no new control parameters are used in this current study. Natural and stiffness controllers are not available with linear error equation. There is no particular way to select the control gain K_{pn} and K_{vn} but according to the proof of stability, $K_p > 0$ and $K_v > 0$ are the conditions for convergence. To select the values of control gain K_p and K_v , the matrix M_h can be used as indications. The matrix M_h contains the information of finger and object inertia. The values of K_{pn} and K_{vn} are found as 60 and 20 by trial and error. In Figure 5.16, the overshoots of z_o and y_o are found as 0.01% and 0.3% respectively. The overshoots from ϕ_o and ψ_o are between 0.01 – 1%.

For the stiffness control law, the only change is the introduction of an adaptation parameter λ . The value of λ is chosen by trial as 0.01. The control gains are applied same as the natural controller gains. Therefore, the gains are, $K_{ps} = K_{pn}$ and $K_{vs} = K_{vn}$. Two sets of desired trajectories are traversed for positions of the object but orientation values remained unchanged. In the first case, the desired position $z_o = 0.01$, $y_o = 0.1710$ and in the second case, $z_o = -0.03$, $y_o = 0.1714$. The Figure 5.17 and 5.18 present the object position tracking responses for the case 1 and 2 respectively. Settling times of the object responses are found between 1 – 2%. No overshoots are found in the object position coordinate. with stiffness controller. The orientation angle ϕ_o and ψ_o are improved in terms of overshoot, tracking error and settling time. The angle θ_o varies due to the gravitational effect during grasp.

The motion errors found from the three controllers are compared in Figure 5.19. The tracking errors obtained with natural and stiffness controllers have less tracking errors when the object

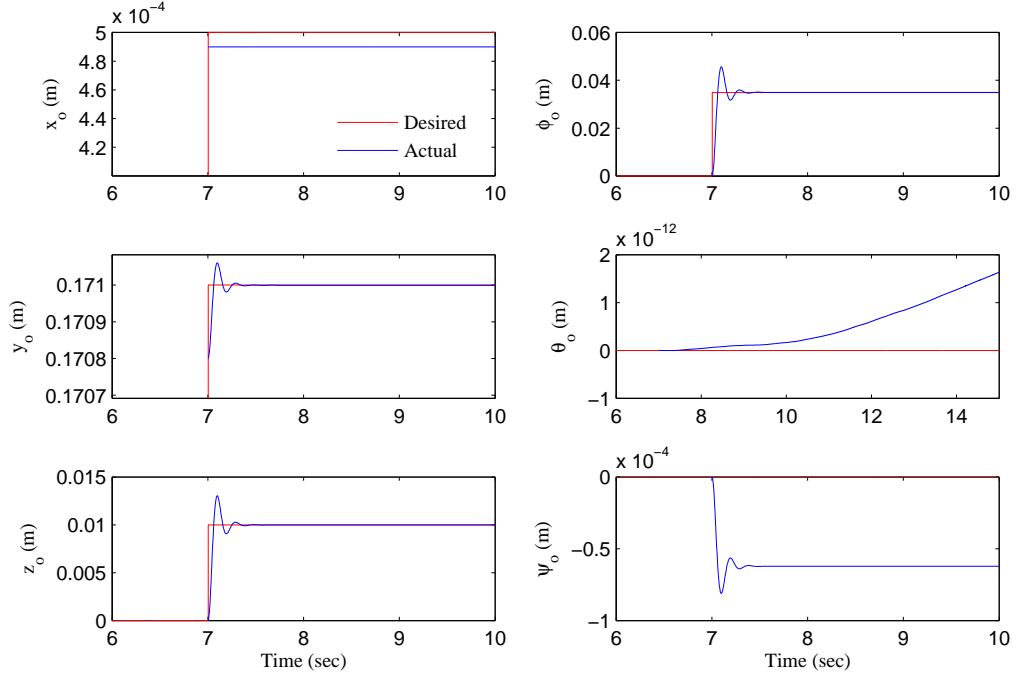


Figure 5.16.: Position tracking of the object after grasped by the Barrett hand with natural controller.

is moved to a desired after grasp. In both position and orientation coordinates, CTC based motions (red dashed lines) tracking errors are larger than the other two. The settling time, overshoots are reduced with natural and stiffness based controllers. The maximum and minimum values of the overshoot, tracking error and settling time for all three (CTC, natural and stiffness) controllers are summarised in the Table 5.4.

Table 5.4.: Range of settling time, tracking error and torque profiles of the joints and the object (before and after grasp).

Controller type	Overshoot (%)	Tracking error (m)	Settling time (second)
CTC	0.5 – 1.1	0.002 – 0.05	1 – 3
Natural	0.01 – 1	0.001 – 0.02	1 – 1.5
Stiffness	0.001 – 0.1	0.001 – 0.01	1 – 1.4

The results from the Table 5.4 depicts that stiffness controller based object manipulation have almost no overshoot in the motion responses. The oscillation in orientation is greatly reduced with stiffness controller. The tracking and settling time are less compared to the results achieved from CTC and natural controllers. The performances of natural and stiffness controllers are similar in terms of the response characteristics. The CTC based simulation appears to have more overshoots, tracking error and settling time compared to the remaining two con-

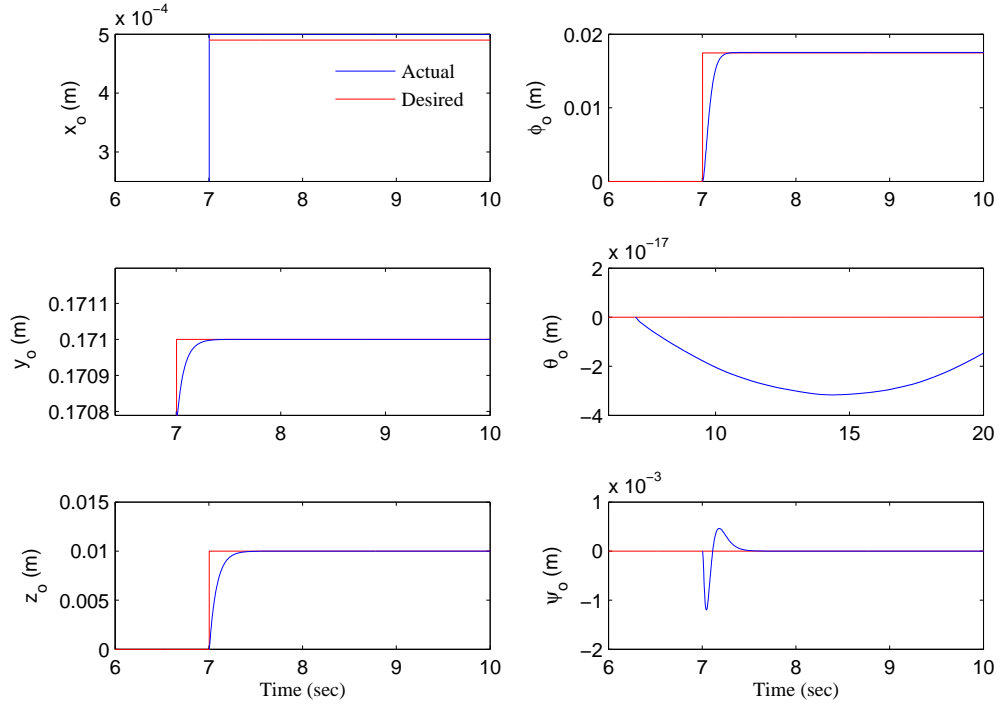


Figure 5.17.: Object position tracking after grasped by the Barrett hand with the stiffness controller (case 1).

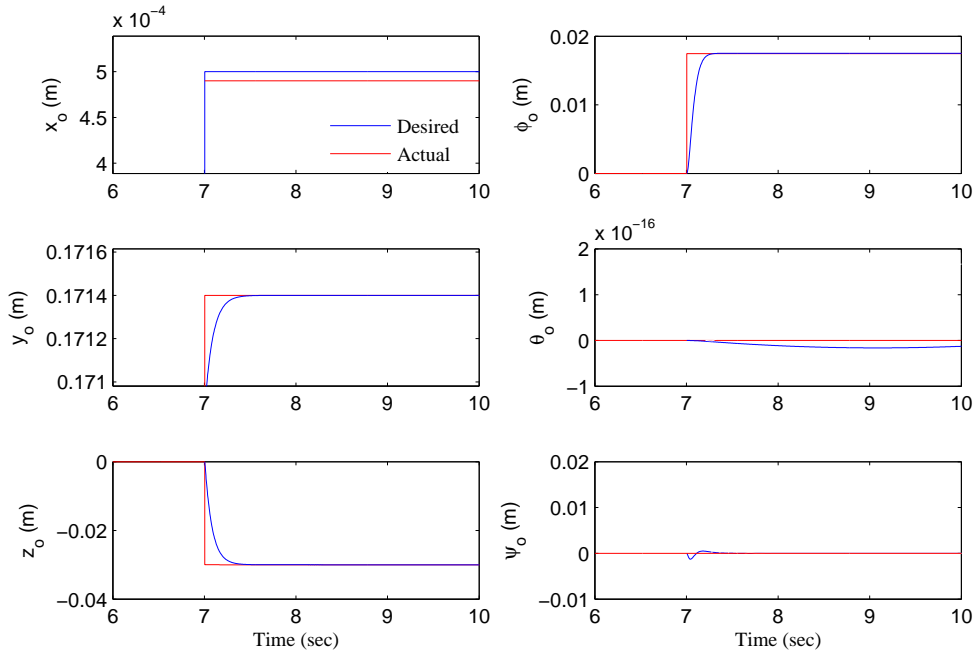


Figure 5.18.: Object position tracking after grasped by the Barrett hand with the stiffness controller (case 2).

trollers. From this analysis, the natural and stiffness controller exhibits better characteristics and proved to be more efficient in achieving tracking performances than CTC for grasping and manipulation by the Barrett hand.

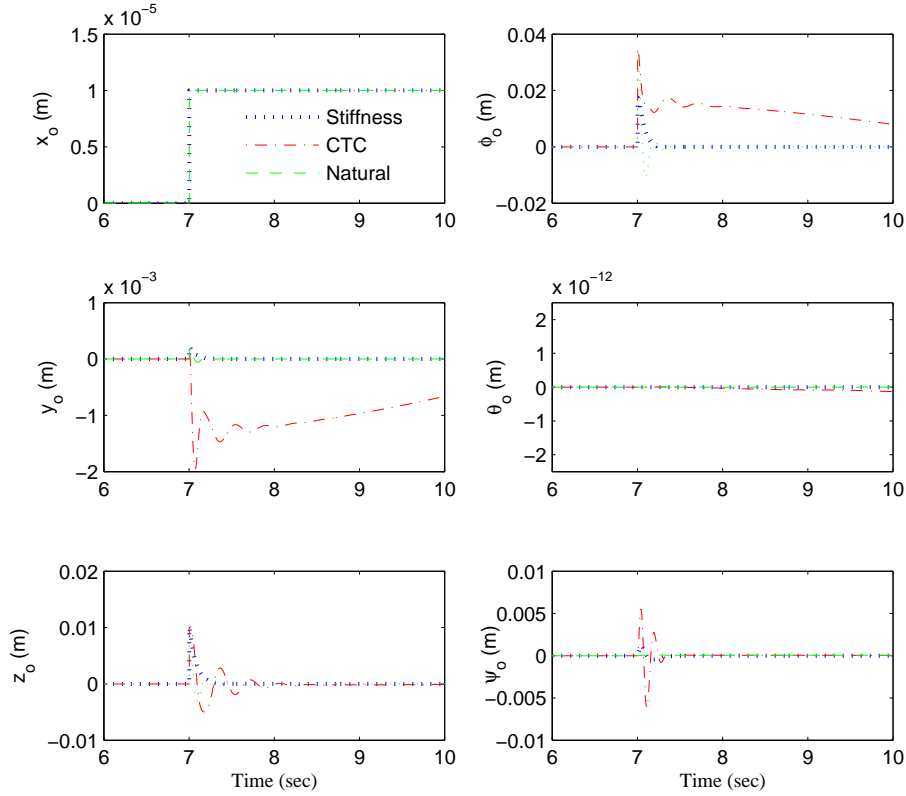


Figure 5.19.: Comparison of object motion based on CTC, Natural and Stiffness control algorithm.

In this section, the grasp model developed between the Barrett hand and an object considering the tangential friction and grasp constraint is presented. For implementing grasping and manipulation controller, grasp map and hand Jacobian are derived for the Barrett hand-object dynamics. The grasp constraint is applied to transform the hand dynamics into the object space. This implies that the control law needs to be derived in the object frame. Using the integrated robot-object dynamics model, three classic controllers are developed to assess the performance of the grasp model. The tracking results of the object position and the orientation are achieved and analysed with the CTC, Natural and stiffness based controllers. Natural laws improved the position errors compared to CTC where the stiffness algorithm reduces oscillation over positions and orientations. The performance of the stiffness based controller inspired the implementation of the adaptive controller presented in the next section.

5.6. SMC-adaptive controller for grasping and manipulation

Feedback linearisation based controllers are not able to compensate the uncertainties in grasping and manipulation such as unknown internal mechanism of the robot hand, joint frictions, contact model etc. Precise position tracking is necessary for coordinated based control. Online estimations of dynamic parameters are required for varying payload during the manipulation tasks. The contact force need to be controlled during grasp not to cause any damage to the object during the grasp. The SMC is a nonlinear robust control method (discussed in chapter 4). The SMC based controller steers the system state (including nonlinearities and uncertainties) to the equilibrium with high control frequency. Hence, robot system with nonlinearities and uncertainties are dealt with SMC [86]. In the SMC method, the open loop response of the dynamic system is modelled with ordinary differential equations. This is related to the variable structure control due to its discontinuous control nature although the control input is expressed in the continuous domain. This method is conceived to restrain the system state trajectory on a chosen sliding surface in the phase space. The Lyapunov approach is used to formulate the SMC controller. The controller guarantees the robustness regardless the presence of uncertainties in a system [86]. This also provides fast dynamic responses, disturbance rejection, parameter variation. The SMC controller with a PID sliding surface was used in manipulator systems in the past [123],[124].

When a object is grasped by the multi-fingered robotic hand, it is assumed that the exact model of the constrained dynamics is known. In this case, parameters of the system such as gravitational loads differ and are not known precisely in advance. Adaptive constrained motion based control is effective to deal with unknown load parameters in different tasks. All adaptive based constraint motion control are based on the reduced dynamic model [125]. In the literatures, modelling and coordinated control based on the hybrid method are proposed assuming that the dynamic parameters of the robotic hand and the object dynamic parameters are well known [105, 106]. Grasping and manipulation tasks become difficult without knowing these parameters. To get rid of these dependencies of system parameters for varying load, adaptation is effectual in achieving the desired motion of the object. By adaptation, both robot and object dynamic parameters are estimated adaptively without knowing any previous knowledge of the two systems.

Grasping and manipulation control objectives refers to the position and the compliant based force control discussed at the beginning of this chapter. The position control is considered as trajectory tracking in the joint or the Cartesian space. Coordinated control is understood by all fingers of a robotic hand for reaching the contact location on the object surface at the same time. Accurate tracking of the multi-fingered hand in the Cartesian space is necessary to meet the requirements of coordinated control. In this section, the SMC based adaptive controller is implemented for grasping and manipulation with the multi-fingered Barrett hand. The SMC

controller part deals with the uncertainties and dynamic non-linearities. The adaptive controller contributes in estimating the unknown dynamic parameters with payload.

5.6.1. Hand-object dynamics for SMC-adaptive controller

The reference motion of the hand is generated by adaptively estimated object reference dynamics. Adaptation facilitates to determine the exact torques are required by robot joints to grasp the object when considering contact dynamics. The sliding function is chosen in higher order for adaptive control. The hand dynamics from equation (2.23) and object dynamics from equation (2.49) are combined to represent the coupled hand-object dynamics based on contact constraint. The adaptive controller is designed and implemented to the hand-object dynamics. The hand dynamics are rearranged for adaptation as follows,

$$M_F \ddot{q} + C_F(q, \dot{q}) + G_F(q) = Y_F(q, \dot{q}, \ddot{q}) a_F, \quad (5.39)$$

where $Y_F \in \mathbb{R}^{8 \times n}$ is the linear regressor matrix subject to dynamic parameters (n is the number of dynamic hand parameters). a_F is the adaptive parameter to estimate dynamic parameters of the hand. The hand dynamics subject to contact forces become,

$$M_F \ddot{q} + C_F(q, \dot{q}) + G_F(q) = \tau - J^T F. \quad (5.40)$$

Similarly, the object dynamics are rearranged for adaptation as below,

$$M_o \ddot{X} + C_o(X_o, \dot{X}_o) \dot{X}_o = Y_o(X_o, \dot{X}_o, \ddot{X}_o) a_o, \quad (5.41)$$

where $Y_o \in \mathbb{R}^{7 \times n}$ is the linear regressor matrix subject to dynamic parameters (n is the number of dynamic object parameters). a_o is the adaptive parameter to determine dynamic parameters of the object. Now, recall the object dynamics from (2.49) as follows,

$$\tilde{M}(q, x) \ddot{X}_o + \tilde{C}(X_o, \dot{X}_o) \dot{X}_o + \tilde{N}(X_o, \dot{X}_o) = G J_h^{-T} \tau, \quad (5.42)$$

where

$$\begin{aligned} \tilde{M} &= M_o + G J_h^{-T} M_f J_h^{-1} G^T \\ \tilde{C} &= C_o + G J_h^{-T} (C_f J_h^{-1} G^T + M_f \frac{d}{dt} (J_h^{-1} G^T)) \\ \tilde{N} &= N_o + G J_h^{-T} N_f \end{aligned}$$

5.6.2. The SMC-adaptive controller implementation

Control architecture for grasping and manipulation including the position and force control as stated in the chapter 5.6. The CTC, natural and stiffness based controllers designed to track the

position robot fingertip or the object. In this section, controllers are designed considering the contact force. Recall equation (5.28) in order to analyse the force tracking objective as,

$$\lim_{t \rightarrow \infty} (F_{dc} - F_c) = 0 \quad (5.43)$$

which states that, the actual contact force $F_c \in \mathbb{R}^9$ should follow the desired contact force $F_{dc} \in \mathbb{R}^9$ and lie within the friction cone f_c at all times. The actual contact force F_c appears as a function of fingertip and contact position on object surface. Contact force F_c depends on the contact type and stiffness of the object. It is not easy to determine F_d when object stiffness properties are assumed to be unknown. In this situation, the adaptation method proved to be very useful to estimate desired contact force. if the object model can be estimated, then the desired external force F_{do} exerted on the object can be calculated. This external force is used to compute the desired contact force F_{dc} online using the grasp properties stated in the chapter 5.3 as follows:

$$M_o \ddot{X} + C_o(\dot{X}_o, \dot{X}_o) \dot{X}_o = F_{do} = G f_c \quad (5.44)$$

$$f_{dc} = G^+ F_{do} + (I - G^+ G) f_{doi}, \quad (5.45)$$

where G^+ is the pseudo inverse of G and is given by $G^+ = G^T (G G^T)^{-1}$ and f_{doi} is the demand force input. It is seen from () that, desired contact force f_{dc} is easily calculated using the grasp map G and the external force F_o achieved from the reference object model. The implementation steps for the adaptive control are stated below,

1. Desired object trajectories $X_o, \dot{X}_o, \ddot{X}_o$ are used to estimate the external object force F_{do} from estimated object reference model.
2. The desired contact force f_{dc} are calculated from the external force F_{do} .
3. Adaptive sliding based control law is calculated for the robotic hand from the estimated hand model.

5.6.3. Desired external and contact force calculation

The expression of reference velocity of the object is similar to the joint reference velocity $\dot{q}_r = \dot{q}_d - \lambda e$. The reference object velocity v_{ro} is defined as,

$$v_{ro} = T_o(\dot{X}_{do} - \rho e_o), \quad (5.46)$$

where the term T_o is the transformation matrix between the object velocity \dot{X}_o and the object velocity using angular velocity ω_o is recalled from (2.43) as,

$$T_o(X_o) = \begin{bmatrix} I_3 & 0 \\ 0 & T_r \end{bmatrix}, \quad (5.47)$$

where the matrix T_r is the orientation based transformation that is presented as,

$$T_r = \begin{bmatrix} 0 & -\sin\phi & \cos\phi\sin\theta \\ 0 & \cos\phi & \sin\phi\sin\theta \\ 1 & 0 & \cos\theta \end{bmatrix}. \quad (5.48)$$

The object position error $e_o = X_o - X_{do}$ and ρ is a positive constant value. Now, using the reference object velocity v_{ro} , the desired external force F_{do} is calculated as,

$$F_{do} = \hat{M}_o \dot{v}_{ro} + \hat{C}_o v_{ro} - k_o s_o, \quad (5.49)$$

where the terms $\hat{M}_o \dot{v}_{ro} + \hat{C}_o v_{ro}$ are defined as reference object model and is replaced by the linear regressor matrix form given in (5.44) as.

$$Y_o(X_o, \dot{X}_o, v_{ro}, \dot{v}_{ro}) \hat{\sigma}_o = \hat{M}_o \dot{v}_{ro} + \hat{C}_o v_{ro}. \quad (5.50)$$

From (5.49) and (5.50), the force F_{do} is found as,

$$F_{do} = Y_o(X_o, \dot{X}_o, v_{ro}, \dot{v}_{ro}) \hat{\sigma}_o - k_o s_o. \quad (5.51)$$

The object motion sliding error is expressed as s_o and the control gain is k_o . $\hat{\sigma}_o$ is the estimation parameter of the object. Adaptive law to estimate the dynamic parameters of the object is given by,

$$\dot{\hat{\sigma}}_o = \Gamma_o^{-1} Y_o^T (X_o, \dot{X}_o, v_{ro}, \dot{v}_{ro}) s_o, \quad (5.52)$$

where Γ_o is the positive adaptation gain matrix. Substitute (5.42) in (5.37) yields,

$$F_{dc} = G^+ (Y_o(X_o, \dot{X}_o, v_{ro}, \dot{v}_{ro}) \hat{\sigma}_o - k_o s_o) + (I - G^+ G) f_{doi}. \quad (5.53)$$

Equation (5.53) is used to determine the desired contact force which is a function of object motion and velocities. In this case, adaptive law estimates the object dynamic parameters for varying contact stiffness and calculates the desired force depending on the exerted force on the object by the robotic hand. The Jacobian constraint from (5.19) as,

$$G_i^T \dot{x}_o = J_{hi} \dot{q}_i. \quad (5.54)$$

The angular velocities of the hand fingers are calculated from (5.54) as,

$$\dot{q}_i = J^{-1} G^T v_o \quad (5.55)$$

and the desired angular velocities of the fingers are expressed similarly as,

$$\dot{q} = J^{-1} G^T v_{do}. \quad (5.56)$$

The reference angular velocities are defined considering the contact force error e_{fc} and the integrated error η as,

$$\dot{q}_r = J^{-1} (G^T v_{do} + c_o e_{fc} + h_o \eta), \quad (5.57)$$

where the terms c_o and h_o are positive gain matrix, contact force error $e_{fc} = F_{dc} - F_c$ and $\eta = \int_0^t e_{fc} dt$. The sliding based adaptive control law for each finger of the robotic hand is now defined as,

$$\tau_f = Y_f(q, \dot{q}, \ddot{q}_r) \hat{\sigma}_f + J^T F_{dc} - k_f s_f. \quad (5.58)$$

The control law (5.58) is composed of a linear regressor matrix Y_f and an estimation parameter $\hat{\sigma}_f$. The feedback gain matrix is defined as k_f and the joint velocity based sliding error is s_f which has the following expression,

$$s_f = \dot{q} - \dot{q}_r \quad (5.59)$$

The only change is seen in the reference joint velocity \dot{q}_r which combines the force error e_f along with reference object velocity v_{or} . The estimation parameter $\hat{\sigma}_f$ is found from the following adaptive law,

$$\dot{\hat{\sigma}}_f = \Gamma_f^{-1} Y_f^T(q, \dot{q}, \ddot{q}_r) s_f. \quad (5.60)$$

The control torque τ_f is applied at each fingertip due to the normal contact force produced at each contact point. The complete control structure for the adaptive control based grasping and manipulation is given in Figure 5.20.

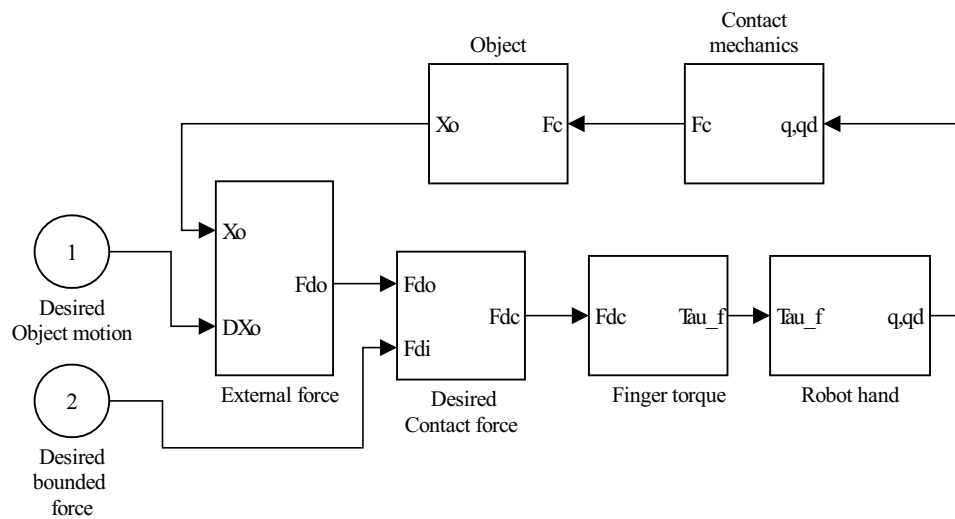


Figure 5.20.: General control structure in workspace before grasping by The Barrett Hand.

5.6.4. Stability of the adaptive controller

Following the grasping and manipulation control problem defined in chapter 5, the control law (5.58) and the adaptive law (5.60) are proposed for the Barrett hand-object based on the complaint dynamics formulated in (5.42). These control laws satisfy the closed loop dynamics such that,

- The actual trajectory X_o, \dot{X}_o converges to the desired one X_d, \dot{X}_d as time $t \rightarrow \infty$.
- And the actual contact force F_c converges to desired contact force F_{dc} as $t \rightarrow \infty$.

Considering both the controllers and the system dynamics subject to the sliding error s , the following expressions are obtained,

$$Y_o(X_o, \dot{X}_o, v_{ro}, \dot{v}_{ro}) \Delta \sigma_o - \Delta F_o - k_o s_o = M_o \dot{s}_o + C_o s_o \quad (5.61)$$

$$Y_f(q, \dot{q}, \dot{q}_r, \ddot{q}_r) \Delta \sigma_f + J^T G e_{fc} - k_f s_f = M_f \dot{s}_f + C_f s_f, \quad (5.62)$$

where $\Delta \sigma_o$ and $\Delta \sigma_f$ are the error vectors of estimated object and finger dynamics parameters respectively. The error vector of the external force is found as ΔF_o . Now, consider the Lyapunov function candidate as,

$$V = 1/2 \left(s_o^T M_o s_o + \Delta \sigma_o^T \Gamma_o^{-1} \Delta \sigma_o + s_f^T M_f s_f + \Delta \sigma_f^T \Gamma_f^{-1} \Delta \sigma_f + \eta^T h_o \eta \right). \quad (5.63)$$

The time derivative of the equation (5.63) gives the solutions as,

$$\begin{aligned} \dot{V} = & \left[s_o^T (M_o \dot{s}_o + 1/2 \dot{M}_o s_o) + \Delta \dot{\sigma}_o^T \Gamma_o^{-1} \Delta \sigma_o \right. \\ & \left. + s_f^T (M_f \dot{s}_f + 1/2 \dot{M}_f s_f) + \Delta \dot{\sigma}_f^T \Gamma_f^{-1} \Delta \sigma_f + \dot{\eta}^T h_o \eta \right]. \end{aligned} \quad (5.64)$$

Now, considering the object matrix properties $(\dot{M}_o - 2C_o)$ and the robotic hand properties $(\dot{M}_f - 2C_f)$, equation (5.42) can be rearranged as,

$$\dot{V} = -s_o^T k_o s_o - s_f^T k_f s_f - e_{fc}^T c_o^T e_{fc} \leq 0. \quad (5.65)$$

In the above equation, both sliding error and estimated error vectors are bounded which implies that V is the Lyapunov candidate for the system. As dynamic parameter σ_f and σ_o are constant values, therefore their estimations are bounded, too. The boundedness of the desired object motion leads the finiteness of the actual motion. Grasp matrix G , hand Jacobian J and desired contact force F_{dc} is bounded. Hence, external object force F_o and actual contact force F_c are also bounded. Now differentiating equation (5.65) yields the form below,

$$\dot{V} = -2 \left(\dot{s}_o^T k_o s_o + \dot{s}_f^T k_f s_f + \dot{e}_{fc}^T c_o^T e_{fc} \right). \quad (5.66)$$

All error based terms on the right side of equation (5.66) are bounded which implies that \dot{V} is bounded. Therefore, time derivative of Lyapunov candidate \dot{V} is continuous. From Lyapunov properties explained in [Slotine, Li], it is proved that $\dot{V} \rightarrow 0$ when $t \rightarrow \infty$. This concludes the convergence of all error and sliding error terms tends to zero when $t \rightarrow \infty$.

5.7. Simulation

The SMC-adaptive controller is applied to the Barrett hand to assess the position and force tracking results when the hand grasps an object. The dynamic parameters of the robotic hand and the object during payload are considered as unknown. The object reference positions are considered as $(x_o, y_o, z_o, \phi_o, \theta_o, \psi_o) = (0.01, 0.1705, 0.01, 0.175, 0.26, 0.055)m$. The grasping and manipulation state is considered with constraints of (5.19). The object mass is considered, $m_o = 1$, the object dimensions are $(a_1, b_1, c_1) = (0.06, 0.05, 0.05)m$, control parameters are obtained by trial and error as, $\rho = 0.1, k_o = 0.05, k_f = 0.0125, c = 0.001, h = 0.001, \tau_f = 0.1, \tau_o = 0.1$. The desired joint angles are considered as $q_{2d} = 71^\circ, q_{3d} = 31^\circ$.

5.8. Results and discussions

Figure 5.21 illustrated the joint motions of the Barrett hand during manipulation. For all joints, the positions are changed when the object is grasped by all fingers. The changes are due to the reaction torque transformed through the contact forces. The joint positions are settled down within 3 seconds of the grasps being made by the hand.

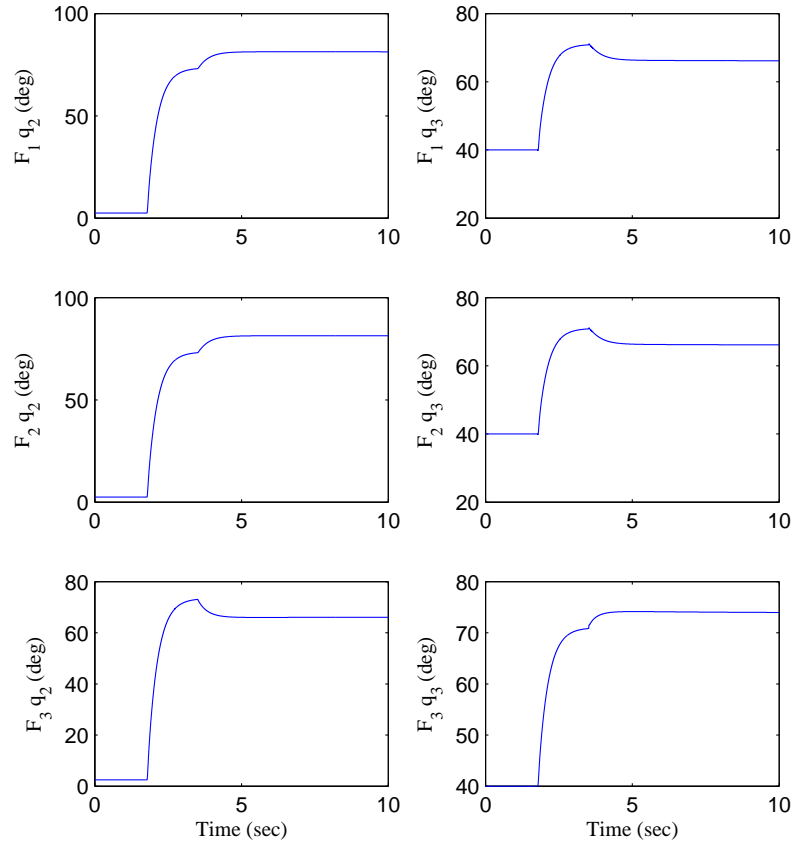


Figure 5.21.: Joint position responses of the Barrett fingers before and after grasp.

In Figure 5.22, the linear motions of the object are shown for case 1. Little overshoots ($< 0.1\%$) are found and settling down times (~ 1 second) are achieved fast for all responses.

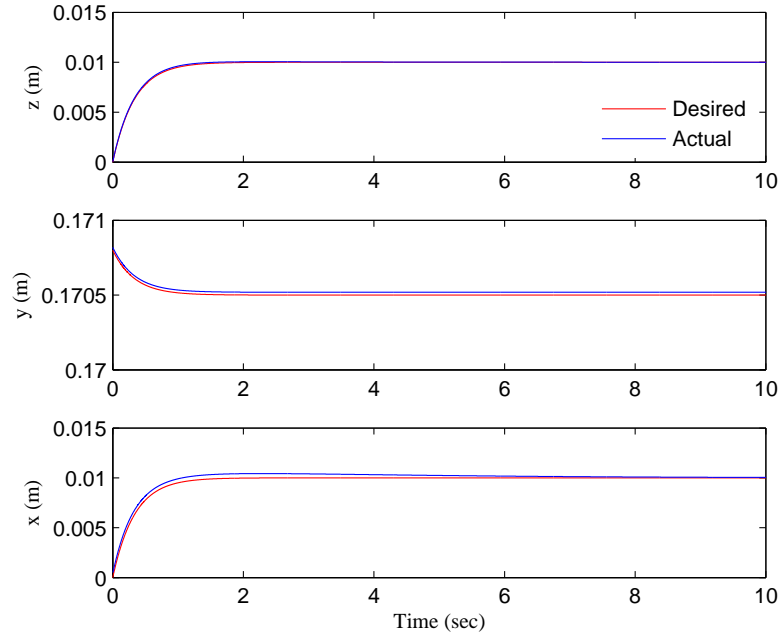


Figure 5.22.: Linear motion of the object after grasp is made by the Barrett hand.

Table 5.5.: Position tracking errors obtained from Figure 5.22.

Tracking error	$x_o(m)$	$y_o(m)$	$z_o(m)$
Linear	0.4 – 0.9	0.1 – 0.2	0.2 – 0.12

The maximum and minimum tracking errors are obtained from the Figure 5.22 and presented in the Table 5.5. The maximum errors are visible in the x_o direction but overall errors are less to obtain the desired position of the object with the SMC-adaptive controller. In Figure 5.23, the linear velocities of the object are presented. Object velocities dissipate energy within short time (< 4 seconds) which implies the smooth manipulation of the object. In Figure 5.24, the orientation tracking results are obtained for the object. The tracking errors are summarised in Table 5.6. The position ϕ_o has errors greater than the other two orientation angles with overdamped responses seen prior to settling down. The maximum error stays within 2% of the final values. Overshoots are found to be less than 1% of the desired angles and settling down times are within 1 second of the starting time.

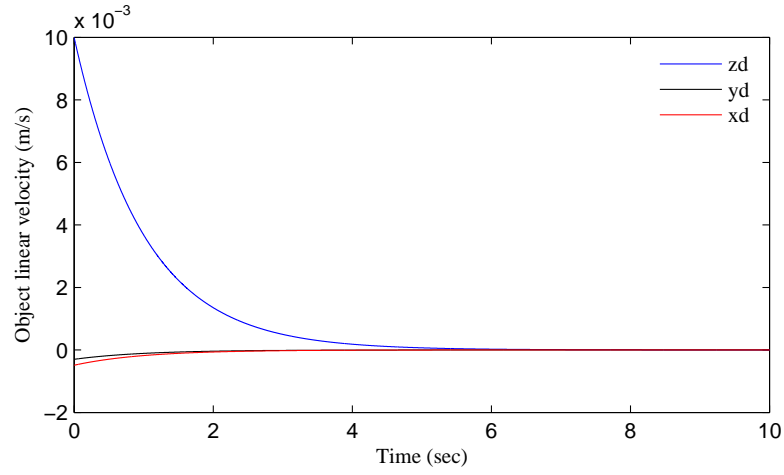


Figure 5.23.: Linear velocity results of the object after grasp is made by the Barrett hand.

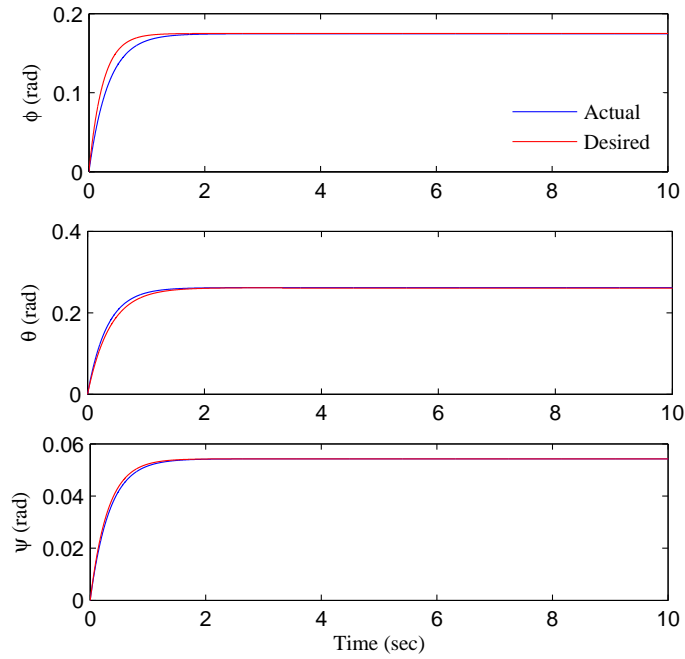


Figure 5.24.: Orientation results of the object after grasp is made by the Barrett hand.

Table 5.6.: Orientation tracking errors obtained from Figure 5.24.

Tracking errors	$\phi_o(\text{rad})$	$\theta_o(\text{rad})$	$\psi_o(\text{rad})$
Orientation	0.05 – 2	0.1 – 0.3	0.2 – 1.4

Figure 5.25 illustrates the mass parameters estimated from the adaptation with F_1 dynamics. All values of the adaptive parameters are settled down at certain values after the initialisation.

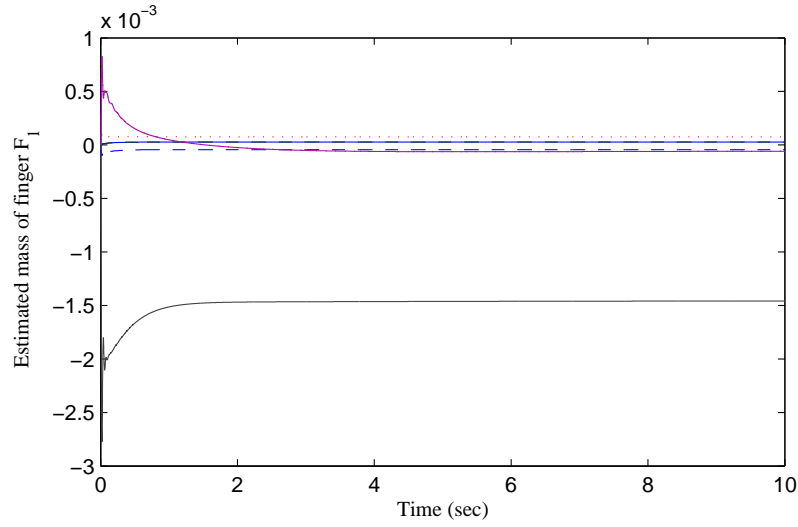


Figure 5.25.: Estimated mass parameters of the integrated dynamics with finger F_1 .

As finger F_1 and F_2 are identical, only mass parameter estimations are obtained of the finger F_3 dynamics in Figure 5.26. In this case, the parameter estimations are found linear and final values are achieved within 1.5 seconds of the initial time.

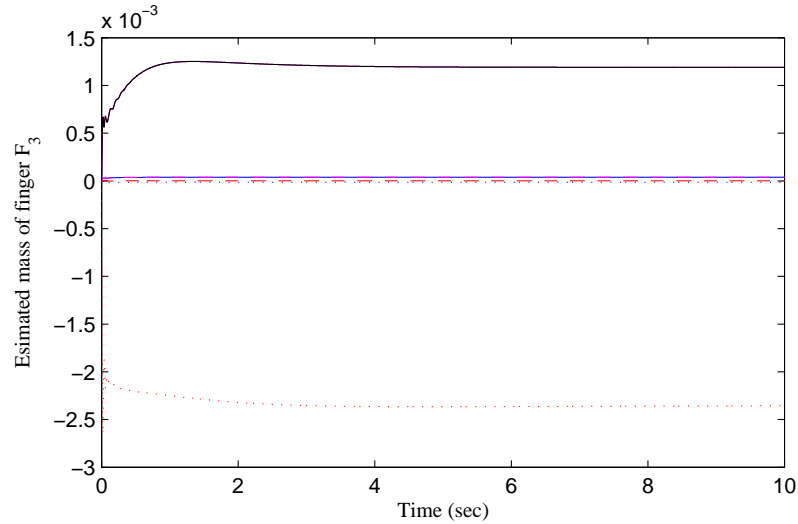


Figure 5.26.: Estimated mass parameters of the integrated dynamics with finger F_3 .

The errors of the contact forces which were obtained through object manipulation by the Barrett hand are shown in the Figure 5.27. For the desired force of $5N$, the actual force errors are found between $0.01 - 0.05N$ ($\sim 1 - 5\%$ of the desired response) after the settling time of 1.11 seconds. This implies that the contact force tracking performance is satisfactory in terms of the error and settling time. In Figure 5.28, the responses of sliding function variables are plotted. Nine sliding functions were generated from three Barrett fingers are plotted in the figure. The

sliding function objective is to reach the sliding surface $s = 0$. The convergence of the sliding variables towards zero is obtained around 6 seconds. In Figure 5.29, the angular velocities of the object are shown. The orientation velocity variables ϕ_d, θ_d, ψ_d dissipates energy and tend towards zero within 2 seconds of the object grasped.

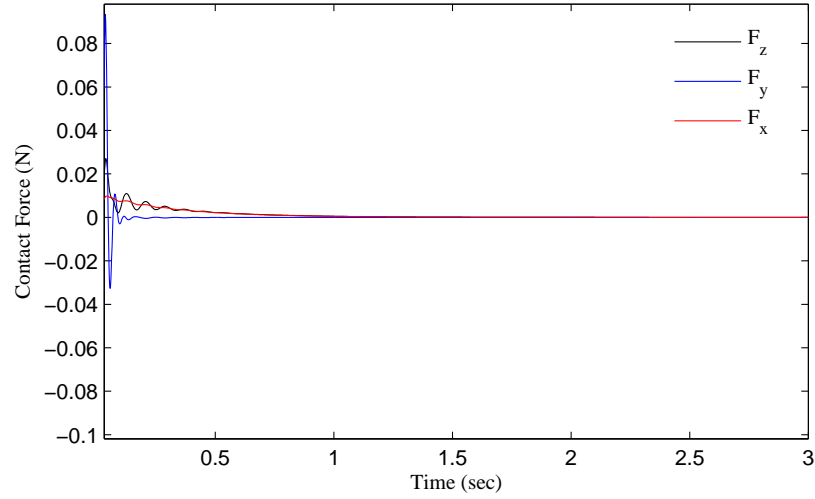


Figure 5.27.: Contact force errors after the object grasped by the hand.

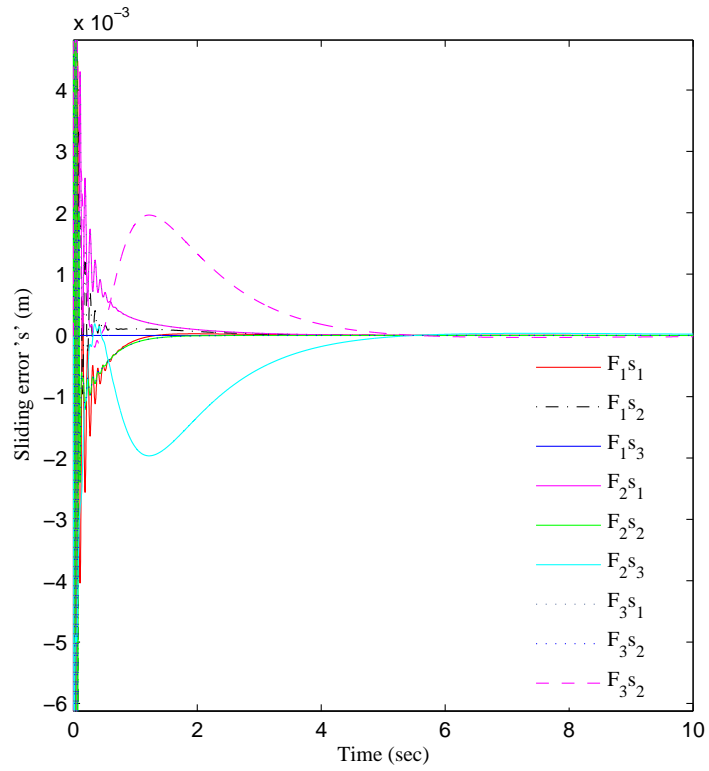


Figure 5.28.: Sliding function errors of Barrett fingers.

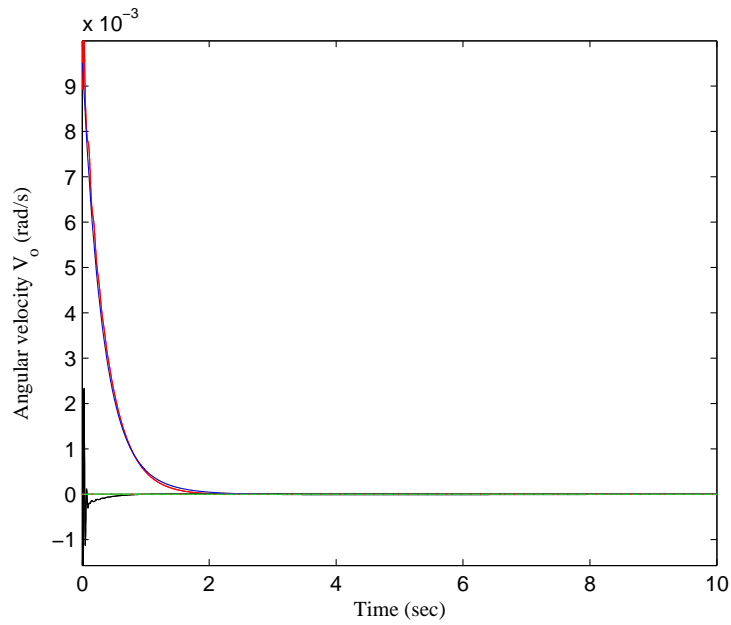


Figure 5.29.: Angular velocities of the object after grasp.

The results obtained from 5.21 – 5.29 are summarised in the Table 5.7.

Table 5.7.: Result analysis of object motion, contact forces, parameter estimation and sliding errors.

Response profile	Summary
Object motion	Motion errors are obtained $\sim 2\%$ of the final position. The positions exhibits less than 1% overshoot and settling times are achieved within 1 second of the initial response.
Contact forces	Errors in contact force between 1 – 5% of the desired force when considering estimated object dynamics both with the adaptive controller and the contact constraint. Settling times are found within 1.1 seconds after the grasp.
Parameter estimation	The parameters are estimated for the robot and the object dynamics. In both cases, final steady values of all parameters are obtained within ~ 2 seconds of the initial time.
Sliding error	The sliding errors $s \approx 0$ are obtained within 6 seconds of the starting time. The $s \approx 0$ convergence implies that the robot-object state trajectory stays in close proximity of equilibrium shortly after the dynamic parameters are estimated.

According to the analysis of Table 5.7, the proposed and implemented SMC-adaptive controller of (5.58) including (5.59) and (5.60) are found suitable for object grasping and manipulation by the multi-fingered Barrett hand. The actual motion and forces response errors are achieved within $\sim 2 - 3\%$ of the desired object and the contact force values.

In this section, SMC-adaptive control method is proposed and implemented for grasping and manipulation of an object by multi-fingered robotic hand. Analysis of joint and object motion tracking results show the efficiency of the controller in tracking of the object position and the contact forces. The adaptation is applied to estimate the robot and object dynamics online during the manipulation. The addition of adaptation contributes to help the hand achieve the desired force for grasping with less error. The position and force tracking ability of this controller with adaptation have the potential to be implemented in practical grasping and manipulation scenario. In real cases, adaptations in controllers greatly improve the tracking performances where there is an unknown internal mechanism for the robot hand and object mass properties.

5.8.1. Comparison of the controllers

In this chapter, four different controllers are proposed and implemented for grasping and manipulation of an object by the Barrett hand. The classical CTC methods are first implemented for the hand. The natural and stiffness controllers are proposed to improve the performance of the object position tracking. These controllers are unable to track the contact force during grasp and manipulation. The new SMC-adaptive controllers are proposed and implemented not

only to track the object position, but also the contact forces after the grasp. The features of all the controllers and performances are summarised in Table 5.8.

Table 5.8.: Comparison of the implemented controllers for grasping and manipulation by the Barrett hand.

Controller name	Features and performance
CTC	This controller cancels the non linear dynamics of the robot-object system and linearises the system. A linear control law is then chosen to meet the control objective with no force tracking features. Tracking performance is achieved within $\sim 5\%$ of the final values with known mass and dynamic parameters.
Natural and Stiffness	The natural controller does not cancel the non linearities. It depends on the skew symmetric properties of the robot-object dynamics and define a single controller without linearising the dynamics. The stiffness controller is similar to the natural one but has online update features for selecting the control gains. Force tracking features are also not available in this case. Tracking performance is achieved within $\sim 2\%$ of the final values with known mass and dynamic parameters. Overshoots and settling time values are improved compared to the CTC.
SMC adaptive	Hybrid control technique is essential to meet the position and force tracking control objectives for grasping and manipulation of an object by the robot hand. A new SMC- adaptive algorithm is proposed and implemented to track the object position as well as the desired contact forces. The controller is designed to track a desired object position and the contact forces by estimating the robot-object dynamic parameters adaptively. The tracking errors are obtained $\sim 2\%$ of the final values considering unknown dynamic parameters. The contact force errors also stay within the $\sim 5\%$ of the desired contact force. The advantages of this controller is the object position tracking ability for unknown system parameters and it can reach the desired contact forces with less error after the object is grasped.

6. Conclusions and Future Work

This research contributions to the developments in the filed of robotic hands are:

- Improved mathematical modelling of a multifingered Barrett hand,
- Nonlinear robust adaptive controllers for precise joint and fingertip tracking of a robot hand,
- Interactional contact modelling between the robot hand and the object,
- Adaptive force based impedance control for force tracking in an unknown environment,
- Evolution based impedance parameter selection for impedance control system and
- Robust adaptive control framework for grasping and manipulation.

Considering first the motivation behind the research and the details of the contributions, prospects and opportunities for future improvements are discussed in the next section.

In this research, the motivation was to take the control challenges of multifingered grasping and manipulation for various potential robotic hand applications. A number of well-engineered, advanced hand models are currently available in industry without an intelligent control solution for performing object grasping and manipulation. The current controllers available for grasping and manipulation have inadequate robustness against uncertainties. This indicated the opportunity for robust control techniques to be implemented into the multifingered grasping and manipulation control framework. Adaptive control methods were found promising in order to estimate the system parameters for varying payload in grasping and manipulation. On the basis of the object position or the contact force control, either robust or adaptive techniques are considered where applicable to deal with the various uncertainties encountered.

The main contributions of the research are: (1) robust adaptive control framework for grasping and manipulation, (2) adaptive force based impedance control for force tracking in unknown environment, (3) nonlinear robust adaptive controllers for precise joint and fingertip tracking of hand fingers with uncertainties and (4) evolution based impedance parameter selections for impedance control system. There have been other contributions that were made in the process of achieving the aim. The initial approach for developing controllers for a system includes the modelling of the robotic hand. The controller accuracy depends on the efficacy of the model

of the system. A SimMechanics based model of the three fingered Barrett hand is developed using the Lagrangian method. The hand model is employed for testing the controllers, but can be utilised as a general three fingered hand test bed for future grasping and manipulation research activities.

The controller selection and development process for grasping and manipulation was not straightforward due to the robot hand's dynamics uncertainties, contact constraints and unknown object properties. The process started with a CTC based controller and eventually evolved into a robust adaptive controller. A CTC based controller is developed and applied to the Barrett hand model. The results are obtained to assess the joint and hand fingertip tracking performances for arbitrary reference position trajectories with known dynamics. The CTC controller cannot compensate the error originating from the modelling deficiencies when the dynamic parameters are imprecisely known and disturbances are introduced in the system. However, it is very common to expect modelling inaccuracies due to uncertainties in a practical case of grasping and manipulation. The control algorithms with robust or adaptive features are suitable to compensate for uncertainties and can achieve the desired manipulation tasks.

Three HOSM based controllers are proposed and developed considering the dynamic uncertainties of the Barrett hand. These controllers are introduced to reduce the joint tracking errors and the chattering of control torques when uncertainties are present. An adaptive controller is added with HOSM, called HOSMA to estimate the varying hand payload on the hand. Noise is assumed to be present in the robot dynamics to see the variations in tracking errors and control signals. Both HOSM and HOSMA based controllers were tested with planar and Barrett fingers. The maximum tracking errors of the Barrett finger in the presence of noise are greatly reduced with HOSM (8%) and HOSMA (5% compared to SMC (15%). The chattering is also reduced with HOSM and HOSMA. The results obtained from the tests of the closed loop system show that the controllers are capable of dealing with uncertainties and have the potential to be applied in any type of multifingered hand to reduce tracking variation and control chattering.

The contact model between Barrett fingers and an object is developed to determine contact forces during manipulation. Position and force based impedance controllers are implemented to minimise the contact forces. The position based impedance controller contributes to minimise the contact forces whereas the force based impedance controllers have the features required to generate a desired force for tracking. In both cases, contact forces were optimised to be in the range of 4 – 8N although the impedance based controller has limitations for real applications in order to deal with the uncertainties.

An adaptive force based impedance controller is developed to track desired contact forces and estimate the stiffness values of the object in real-time, during contact. The results obtained from this controller illustrates that it is able to track contact forces for any interaction without knowing the stiffness values of the object. The interaction is very common in multifingered grasping and manipulation. Integration of adaptation with an impedance controller would not

required the specification of the stiffness values for robot-object interaction. The crucial tasks in impedance based control are the selection of impedance parameters which will describe the coupled robot-object system's transient behaviour similar to the characteristics of second order mass spring damper system. There are no straightforward methods available to obtain these parameters rather than by trial and error.

To avoid the time consuming parameter tuning process, an evolutionary algorithm called CMS-EA is used with the data of the contact forces collected from the impedance controller. The CMS-EA algorithm searches for the best parameters which gives minimal error for the desired and actual contact forces. The parameters obtained through the searching process are tested in the impedance based control environment. The contact force results are tracked with less overshoot ($\sim 1 - 3\%$) for different desired force input. The impedance controller is widely used for controlling the interaction between the two bodies. In an industrial situation, the CMS-EA based algorithm will contribute to finding the best impedance parameters for achieving force tracking accuracies. The future step is to develop the real-time estimation process for determining the impedance parameters.

When an object is grasped for manipulation, the contact forces also need to be controlled as well as the position of the object. This situation brings complication in developing the controllers because both object position and contact force need to be controlled at the same time. Three position based controllers are formulated and applied in the case of Barrett hand grasping an object. The manipulation is performed in the object space to achieve the desired object motion (position and orientation). Amongst the controllers, the stiffness controller is found with maximum 1% of object tracking errors where the control gains were adapted to minimise the error. The CTC and natural controller based manipulation show a maximum of 5% and 3% of maximum error respectively. These controllers have no features to control the contact force and are also unable to deal with unknown robot or object dynamics and hence they are not feasible in practical applications. Model mismatching due to uncertainties is common amongst most systems from a practical point of view. A SMC-adaptive controller is developed which is able to control the object motion and the contact forces at the same time by estimating the unknown robot and object dynamics. The feature of the developed controller is its real-time estimation ability of the robot and object dynamic parameters when it is grasped. The controller does not depend on mass or stiffness parameters for accomplishing the manipulation. From the results and analysis of the previous chapter, the SMC-adaptive controller is able to achieve any desired object position and orientation and to meet the reference contact force requirements. The SMC-adaptive controller is developed in generalised format and can be applied for grasping and manipulation with other hand models. The parameter free estimation features make it suitable for applying to real grasping and manipulation. This would be the immediate next step of the research to implement the controller in real world grasping and manipulation by the robot hand.

6.0.2. Future directions

The future developments on the current multifingered hand will lead to implement a more advanced control framework for performing higher level manipulation tasks such as: determining the optimal grasp for a given object, visual control tasks for object localisation, and a logic based scheduling of a full manipulation sequence. Optimal control techniques have stability and robustness characteristics to deal with the force optimisation problem. The control objective is first specified in this technique and a cost function is established based on the control objective. The control goal is to minimise this cost function by adaptive techniques. Another popular technique is Model Predictive Control (MPC) which predicts the future output state at each instant. This predicted output is computed on the basis of the past input, the output and the future control signals. It has significant advantages for dealing with the large number of manipulated and controlled variables. Also, the grasping constraints imposed on the robot-object dynamics can be dealt with MPC techniques. This technique suggests computing the control signals optimally according to the control formulation and while ensuring the output is very close to the desired values. The objective function is developed based on the errors between the predicted output and the predicted desired trajectory. The control law is developed according to the estimation of the objective function.

Fuzzy logic based controller is another popular method widely used in robotics. In this work, there were intentions from the beginning to develop a Fuzzy based controllers for the Barrett Hand which has not been possible due to time constraints. The initial objective was to develop a fuzzy sliding based adaptive controller for the grasping and manipulation of an object by the Barrett hand. However the complete incorporation of these techniques was not possible in the time frame of the project and will be considered in the immediate future. Moreover, the evolution and the adaptation used in Chapter 5 for implementing impedance controller can be considered to extend the capabilities of the grasping and manipulation controller of the multifingered robot hand. In this case, the real-time impedance parameter selection based adaptive impedance controller will be developed. The controller will mainly control the contact forces by estimating the desired impedance in real-time for the robot-object system and the robot fingertip position will be measured from that. The object position will be calculated from the fingertip position by contact constraint.

The controllers developed in this research provide clarity about the implementation at all levels of grasping and manipulation. These implementations establish a base for future developments with further integration on the robot hands currently available. Referring to the case of the multifingered hand, it is important to notice that the results obtained so far, make it possible to perform more natural grasping and manipulation tasks in the context of the human hand.

A. Appendix

A.1. Stability of the SMC controller

The SMC described in chapter 4 is a powerful nonlinear robust control approach to solve the uncertain control problem in nonlinear systems. In SMC, this idea is developed where a sliding surface is designed such that the system trajectory remains along the sliding surface by a discontinuous law. The system dynamics is controlled by the sliding surface variable and independent of the system variables. The control goal is therefore to reach the sliding surface in finite time compensating uncertainties. Considering the control objective of (3.19), the sliding surface is defined as

$$s(t) = \dot{e} + \lambda e = 0 \quad (\text{A.1})$$

where $e = X - X_d$ is the tracking error and λ is a constant whose eigenvalues are strictly positive. The control objective can be achieved by selecting the control input in such a way so that the sliding surface satisfies the condition:

$$\frac{1}{2} \frac{d}{dt} s^2 \leq -\eta |s|$$

where η is a positive constant. The above condition specifies that the energy of the system should decay until it reaches the sliding surface. It can be achieved by replacing the desired position X_d as virtual reference,

$$X_r = X_d - \lambda \int_0^t e dt \quad (\text{A.2})$$

Virtual velocity \dot{X}_r and acceleration \ddot{X}_r therefore can be determined as

$$\dot{X}_r = \dot{X}_d - \lambda e, \quad \ddot{X}_r = \ddot{X}_d - \lambda \dot{e} \quad (\text{A.3})$$

and the sliding surface $s(t)$ can be re-written as,

$$\begin{aligned} s &= \dot{X} - \dot{X}_r \\ \dot{s} &= \ddot{X} - \ddot{X}_r \end{aligned} \quad (\text{A.4})$$

Substituting \dot{s} from (A.4) into the joint space dynamics of (3.2) yields,

$$M(q)\dot{s} = u - C(q, \dot{q})\dot{q} - G(q) - M(q)\ddot{q}_r \quad (\text{A.5})$$

$$M(q)\dot{s} = u - C(q, \dot{q})(s + \dot{q}_r) - G(q) - M(q)\ddot{q}_r$$

Lyapunov's method can be used to prove the stability for the SMC [89]. Consider the Lyapunov function candidate as,

$$V = \frac{1}{2}s^T Ms \quad (\text{A.6})$$

From second properties of chapter 2.4.6, the matrix M is positive definite, $V > 0$ for $s \neq 0$. Therefore, the system is stable when the rate of change of the candidate function $\dot{V} < 0$ according to the Lyapunov stability where,

$$\dot{V} = s^T M \dot{s} + \frac{1}{2}s^T \dot{M}s \quad (\text{A.7})$$

$$\dot{V} = s^T (u - C(q, \dot{q})(s + \dot{q}_r) - G(q) - M(q)\ddot{q}_r) + \frac{1}{2}s^T \dot{M}s$$

Applying the skew symmetric properties, $s^T \dot{M}s = 0$ and $s^T Cs = 0$, the \dot{V} becomes,

$$\dot{V} = s^T (u - M(q)\ddot{q}_r - C(q, \dot{q})\dot{q}_r - G(q))$$

Let the control signal u be given as,

$$u_s = u_a - K^T \text{sign}(s) \quad (\text{A.8})$$

$$u_a = M(q) + C(q, \dot{q}_r)\dot{q}_r + G(q)$$

where $K > 0$ is the switching gain. The control input u yields,

$$\dot{V} = -s^T K^T \text{sign}(s) = -|s|K^T < 0$$

Therefore, the control signal u_s in (A.8) is a stabilizing control law in the sense of Lyapunov.

References

- [1] P. Tuffield and H. Elias, “The shadow robot mimics human actions,” *Industrial Robot: An International Journal*, vol. 30, no. 1, pp. 56–60, 2003.
- [2] J. Tegin, S. Ekvall, D. Kragic, J. Wikander, and B. Iliev, “Demonstration-based learning and control for automatic grasping,” *Intelligent Service Robotics*, vol. 2, no. 1, pp. 23–30, 2009.
- [3] J. Butterfaß, M. Grebenstein, H. Liu, and G. Hirzinger, “Dlr-hand ii: Next generation of a dextrous robot hand,” in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 1, pp. 109–114, IEEE, 2001.
- [4] C. F. Ruoff and J. K. Salisbury Jr, “Multi-fingered robotic hand,” May 1 1990. US Patent 4,921,293.
- [5] W. Townsend, “The barretthand grasper—programmably flexible part handling and assembly,” *Industrial Robot: An International Journal*, vol. 27, no. 3, pp. 181–188, 2000.
- [6] M. Grebenstein, A. Albu-Schäffer, T. Bahls, M. Chalon, O. Eiberger, W. Friedl, R. Gruber, S. Haddadin, U. Hagn, R. Haslinger, *et al.*, “The dlr hand arm system,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3175–3182, IEEE, 2011.
- [7] J. K. Salisbury and J. J. Craig, “Articulated hands force control and kinematic issues,” *The International Journal of Robotics Research*, vol. 1, no. 1, pp. 4–17, 1982.
- [8] T. Yoshikawa and K. Nagai, “Manipulating and grasping forces in manipulation by multifingered robot hands,” *Robotics and Automation, IEEE Transactions on*, vol. 7, no. 1, pp. 67–77, 1991.
- [9] A. Bicchi, C. Melchiorri, and D. Balluchi, “On the mobility and manipulability of general multiple limb robots,” *Robotics and Automation, IEEE Transactions on*, vol. 11, no. 2, pp. 215–228, 1995.
- [10] B. Mishra, J. T. Schwartz, and M. Sharir, “On the existence and synthesis of multifinger positive grips,” *Algorithmica*, vol. 2, no. 1-4, pp. 541–558, 1987.

- [11] M. Zribi, J. Chen, and M. S. Mahmoud, "Coordination and control of multi-fingered robot hands with rolling and sliding contacts," *Journal of Intelligent and Robotic Systems*, vol. 24, no. 2, pp. 125–149, 1999.
- [12] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *ICRA*, pp. 348–353, Citeseer, 2000.
- [13] A. Bicchi, "Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity," *Robotics and Automation, IEEE Transactions on*, vol. 16, no. 6, pp. 652–662, 2000.
- [14] N. Ulrich, R. Paul, and R. Bajcsy, "A medium-complexity compliant end effector," in *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, pp. 434–436, IEEE, 1988.
- [15] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.
- [16] J. K. Salisbury and B. Roth, "Kinematic and force analysis of articulated mechanical hands," *Journal of Mechanical Design*, vol. 105, no. 1, pp. 35–41, 1983.
- [17] J. Kerr and B. Roth, "Analysis of multifingered hands," *The International Journal of Robotics Research*, vol. 4, no. 4, pp. 3–17, 1986.
- [18] M. R. Cutkosky, "Grasping and fine manipulation for automated manufacturing.," *Dissertation Abstracts International Part B: Science and Engineering*[DISS. ABST. INT. PT. B- SCI. & ENG.], vol. 46, no. 6, 1985.
- [19] D. J. Montana, "The kinematics of contact and grasp," *The International Journal of Robotics Research*, vol. 7, no. 3, pp. 17–32, 1988.
- [20] J. C. Trinkle, "The mechanics and planning of enveloping grasps," 1987.
- [21] C. Cai and B. Roth, "On the spatial motion of a rigid body with point contact," in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4, pp. 686–695, IEEE, 1987.
- [22] D. J. Montana, "The condition for contact grasp stability," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pp. 412–417, IEEE, 1991.
- [23] D. J. Montana, "The kinematics of multi-fingered manipulation," *Robotics and Automation, IEEE Transactions on*, vol. 11, no. 4, pp. 491–503, 1995.
- [24] A. B. Cole, J. E. Hauser, and S. S. Sastry, "Kinematics and control of multifingered hands with rolling contact," *Automatic Control, IEEE Transactions on*, vol. 34, no. 4, pp. 398–404, 1989.

- [25] A. A. Cole, P. Hsu, and S. S. Sastry, "Dynamic control of sliding by robot hands for regrasping," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 1, pp. 42–52, 1992.
- [26] F.-T. Cheng and D. E. Orin, "Efficient algorithm for optimal force distribution-the compact-dual lp method," *Robotics and Automation, IEEE Transactions on*, vol. 6, no. 2, pp. 178–187, 1990.
- [27] Y. Nakamura, K. Nagai, and T. Yoshikawa, "Mechanics of coordinative manipulation by multiple robotic mechanisms," in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4, pp. 991–998, IEEE, 1987.
- [28] M. Cutkosky, P. Akella, R. Howe, and I. Kao, "Grasping as a contact sport," in *Proceedings of the 4th international symposium on Robotics Research*, pp. 199–206, MIT Press, 1988.
- [29] R. D. Howe and M. R. Cutkosky, "Practical force-motion models for sliding manipulation," *The International Journal of Robotics Research*, vol. 15, no. 6, pp. 557–572, 1996.
- [30] N. Xydas and I. Kao, "Modeling of contact mechanics and friction limit surfaces for soft fingers in robotics, with experimental results," *The International Journal of Robotics Research*, vol. 18, no. 9, pp. 941–950, 1999.
- [31] Z. Li and S. S. Sastry, "Task-oriented optimal grasping by multifingered robot hands," *Robotics and Automation, IEEE Journal of*, vol. 4, no. 1, pp. 32–44, 1988.
- [32] V.-D. Nguyen, "The synthesis of force-closure grasps," 1985.
- [33] V.-D. Nguyen, "Constructing force-closure grasps," *The International Journal of Robotics Research*, vol. 7, no. 3, pp. 3–16, 1988.
- [34] E. Rimon and J. Burdick, "On force and form closure for multiple finger grasps," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 2, pp. 1795–1800, IEEE, 1996.
- [35] A. Bicchi, Y. Chitour, A. Marigo, and D. Prattichizzo, "Dexterity through rolling: Towards manipulation of unknown objects," in *Proc. of the third International Symposium on Methods and Models in Automation and Robotics*, pp. 879–886, Citeseer, 1999.
- [36] Z. Li, J. Canny, and S. S. Sastry, "On motion planning for dexterous manipulation. i. the problem formulation," in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pp. 775–780, IEEE, 1989.
- [37] J. Hong, G. Lafferriere, B. Mishra, and X. Tan, "Fine manipulation with multifinger hands," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pp. 1568–1573, IEEE, 1990.

- [38] E. Paljug, X. Yun, and V. Kumar, "Control of rolling contacts in multi-arm manipulation," *Robotics and Automation, IEEE Transactions on*, vol. 10, no. 4, pp. 441–452, 1994.
- [39] A. Bicchi and R. Sorrentino, "Dexterous manipulation through rolling," in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 1, pp. 452–457, IEEE, 1995.
- [40] J. K. Salisbury and J. J. Craig, "Articulated hands force control and kinematic issues," *The International Journal of Robotics Research*, vol. 1, no. 1, pp. 4–17, 1982.
- [41] Z. Li, P. Hsu, and S. Sastry, "Grasping and coordinated manipulation by a multifingered robot hand," *The International Journal of Robotics Research*, vol. 8, no. 4, pp. 33–50, 1989.
- [42] R. M. Murray and S. S. Sastry, "Control experiments in planar manipulation and grasping," in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pp. 624–629, IEEE, 1989.
- [43] S. Hayati, "Hybrid position/force control of multi-arm cooperating robots," in *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, vol. 3, pp. 82–89, IEEE, 1986.
- [44] D. Rus, "Dexterous rotations of polyhedra," in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pp. 2758–2763, IEEE, 1992.
- [45] D. Rus, "Coordinated manipulation of objects in a plane," *Algorithmica*, vol. 19, no. 1-2, pp. 129–147, 1997.
- [46] N. Sarkar, X. Yun, and V. Kumar, "Dynamic control of 3-d rolling contacts in two-arm manipulation," *Robotics and Automation, IEEE Transactions on*, vol. 13, no. 3, pp. 364–376, 1997.
- [47] Z. Li, Z. Qin, S. Jiang, and L. Han, "Coordinated motion generation and real-time grasping force control for multifingered manipulation," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 4, pp. 3631–3638, IEEE, 1998.
- [48] C.-Y. Su and Y. Stepanenko, "Adaptive sliding mode coordinated control of multiple robot arms attached to a constrained object," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 25, no. 5, pp. 871–878, 1995.
- [49] H. Kawasaki, T. Shimizu, and S. Ito, "Adaptive coordinated control of multiple robot arms," in *Proceedings of IFAC*, vol. 2, pp. 17–25, Cambridge University Press, 2000.
- [50] P. R. Pagilla and M. Tomizuka, "Adaptive control of two robot arms carrying an unknown object," in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 1, pp. 597–602, IEEE, 1995.

- [51] M. Cherif and K. K. Gupta, "Planning quasi-static fingertip manipulations for reconfiguring objects," *Robotics and Automation, IEEE Transactions on*, vol. 15, no. 5, pp. 837–848, 1999.
- [52] J. Ponce and B. Faverjon, "On computing three-finger force-closure grasps of polygonal objects," *Robotics and Automation, IEEE Transactions on*, vol. 11, no. 6, pp. 868–881, 1995.
- [53] A. Bicchi, "On the closure properties of robotic grasping," *The International Journal of Robotics Research*, vol. 14, no. 4, pp. 319–334, 1995.
- [54] V. Kumar and K. J. Waldron, "Sub-optimal algorithms for force distribution in multi-fingered grippers," in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4, pp. 252–257, IEEE, 1987.
- [55] M. Buss, H. Hashimoto, and J. B. Moore, "Dextrous hand grasping force optimization," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 3, pp. 406–418, 1996.
- [56] M. Buss and K. Kleinmann, "Multifingered grasping experiments using real-time grasping force optimization," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 2, pp. 1807–1812, IEEE, 1996.
- [57] M. Buss and T. Schlegl, "Multifingered regrasping using on-line grasping force optimization," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 2, pp. 998–1003, IEEE, 1997.
- [58] L. Han, J. C. Trinkle, and Z. X. Li, "Grasp analysis as linear matrix inequality problems," *Robotics and Automation, IEEE Transactions on*, vol. 16, no. 6, pp. 663–674, 2000.
- [59] Y. Nakamura, K. Nagai, and T. Yoshikawa, "Dynamics and stability in coordination of multiple robotic mechanisms," *The International Journal of Robotics Research*, vol. 8, no. 2, pp. 44–61, 1989.
- [60] H. K. Khalil and J. Grizzle, *Nonlinear systems*, vol. 3. Prentice hall Upper Saddle River, 2002.
- [61] L. Han, Y.-S. Guan, Z. Li, Q. Shi, and J. C. Trinkle, "Dextrous manipulation with rolling contacts," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 2, pp. 992–997, IEEE, 1997.
- [62] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2008.
- [63] W.-Y. Chung and K. J. Waldron, "Simulations of dexterous manipulation for multifingered systems," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 2321–2326, IEEE, 1994.
- [64] J. Chen and M. Zribi, "Control of multifingered robot hands with rolling and sliding contacts," *The International Journal of Advanced Manufacturing Technology*, vol. 16, no. 1, pp. 71–77, 2000.

- [65] K. Nagai and T. Yoshikawa, "Dynamic manipulation/grasping control of multifingered robot hands," in *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pp. 1027–1033, IEEE, 1993.
- [66] A. DasGupta and H. Hatwal, "Dynamics and nonlinear coordination control of multifingered mechanical hands," *Journal of dynamic systems, measurement, and control*, vol. 120, no. 2, pp. 275–281, 1998.
- [67] A. Isidori, *Nonlinear control systems*, vol. 1. Springer, 1995.
- [68] Z. Doulgeri, J. Fasoulas, and S. Arimoto, "Feedback control for object manipulation by a pair of soft tip fingers," *Robotica*, vol. 20, no. 01, pp. 1–11, 2002.
- [69] C. Remond, V. Perdereau, and M. Drouin, "A hierarchical multi-fingered hand control structure with rolling contact compensation," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 4, pp. 3731–3736, IEEE, 2002.
- [70] R. Hilhorst and K. Tanie, "Dexterous manipulation of objects with unknown parameters by robot hands," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 3098–3103, IEEE, 1994.
- [71] S. Ueki, H. Kawasaki, and T. Mouri, "Adaptive coordinated control of multi-fingered hands with sliding contact," in *SICE-ICASE, 2006. International Joint Conference*, pp. 5893–5898, IEEE, 2006.
- [72] M. R. Cutkosky and I. Kao, "Computing and controlling compliance of a robotic hand," *Robotics and Automation, IEEE Transactions on*, vol. 5, no. 2, pp. 151–165, 1989.
- [73] H. Choi, W. Chung, and Y. Youm, "Stiffness analysis and control of multi-fingered robot hands," *Journal of dynamic systems, measurement, and control*, vol. 117, no. 3, pp. 435–439, 1995.
- [74] J. S. Son and R. D. Howe, "Performance limits and stiffness control of multifingered hands," in *Experimental Robotics IV*, pp. 91–102, Springer, 1997.
- [75] B.-H. Kim, B.-J. Yi, S.-R. Oh, and I. H. Suh, "Independent finger and independent joint-based compliance control of multifingered robot hands," *Robotics and Automation, IEEE Transactions on*, vol. 19, no. 2, pp. 185–199, 2003.
- [76] L. Han, Z. Li, J. C. Trinkle, Z. Qin, and S. Jiang, "The planning and control of robot dextrous manipulation," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 1, pp. 263–269, IEEE, 2000.
- [77] S. A. Schneider and R. H. Cannon Jr, "Object impedance control for cooperative manipulation: Theory and experimental results," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 3, pp. 383–394, 1992.

- [78] S. Stramigioli, C. Melchiorri, and S. Andreotti, "A passivity-based control scheme for robotic grasping and manipulation," in *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, vol. 3, pp. 2951–2956, IEEE, 1999.
- [79] L. Biagiotti, H. Liu, G. Hirzinger, and C. Melchiorri, "Cartesian impedance control for dexterous manipulation," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 4, pp. 3270–3275, IEEE, 2003.
- [80] T. Schlegl, M. Buss, T. Omata, and G. Schmidt, "Fast dextrous re-grasping with optimal contact forces and contact sensor-based impedance control," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 1, pp. 103–108, IEEE, 2001.
- [81] K. B. Shimoga, "Robot grasp synthesis algorithms: A survey," *The International Journal of Robotics Research*, vol. 15, no. 3, pp. 230–266, 1996.
- [82] A. M. Okamura, N. Smaby, and M. R. Cutkosky, "An overview of dexterous manipulation," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 1, pp. 255–262, IEEE, 2000.
- [83] M. W. Walker, D. Kim, and J. Dionise, "Adaptive coordinated motion control of two manipulator arms," in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pp. 1084–1090, IEEE, 1989.
- [84] T. M. Inc., *MATLAB & Simulink*. MATLAB, 2012.
- [85] J. J. Craig, *Introduction to robotics: mechanics and control*. Pearson/Prentice Hall Upper Saddle River, NJ, USA:, 2005.
- [86] J.-J. E. Slotine, W. Li, *et al.*, *Applied nonlinear control*, vol. 199. Prentice-Hall Englewood Cliffs, NJ, 1991.
- [87] G. D. Wood and D. C. Kennedy, "Simulating mechanical systems in simulink with simmechanics," *The Mathworks Report*, 2003.
- [88] F. L. Lewis, C. T. Abdallah, and D. M. Dawson, *Control of robot manipulators*, vol. 236. Macmillan New York, 1993.
- [89] J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *The International Journal of Robotics Research*, vol. 6, no. 3, pp. 49–59, 1987.
- [90] J.-J. Slotine and L. Weiping, "Adaptive manipulator control: A case study," *Automatic Control, IEEE Transactions on*, vol. 33, no. 11, pp. 995–1003, 1988.
- [91] A. Levant, "Sliding order and sliding accuracy in sliding mode control," *International journal of control*, vol. 58, no. 6, pp. 1247–1263, 1993.
- [92] V. I. Utkin, *Sliding modes in control and optimization*, vol. 116. Springer, 1992.

- [93] Q. Zong, Z.-S. Zhao, and J. Zhang, "Higher order sliding mode control with self-tuning law based on integral sliding mode," *IET control theory & applications*, vol. 4, no. 7, pp. 1282–1289, 2010.
- [94] A. Levant, "Higher-order sliding modes, differentiation and output-feedback control," *International journal of Control*, vol. 76, no. 9-10, pp. 924–941, 2003.
- [95] R. Palm, "Sliding mode fuzzy control," in *Fuzzy Systems, 1992., IEEE International Conference on*, pp. 519–526, IEEE, 1992.
- [96] A. Levant and L. Alelishvili, "Integral high-order sliding modes," *IEEE Transactions on Automatic control*, vol. 52, no. 7, p. 1278, 2007.
- [97] S. Laghrouche, F. Plestan, and A. Glumineau, "Higher order sliding mode control based on integral sliding mode," *Automatica*, vol. 43, no. 3, pp. 531–537, 2007.
- [98] F. Dinuzzo and A. Ferrara, "Higher order sliding mode controllers with optimal reaching," *Automatic Control, IEEE Transactions on*, vol. 54, no. 9, pp. 2126–2136, 2009.
- [99] M. Defoort, T. Floquet, A. Kokosy, and W. Perruquetti, "A novel higher order sliding mode control scheme," *Systems & Control Letters*, vol. 58, no. 2, pp. 102–108, 2009.
- [100] G. Bartolini, A. Ferrara, and E. Usai, "Chattering avoidance by second-order sliding mode control," *Automatic control, IEEE Transactions on*, vol. 43, no. 2, pp. 241–246, 1998.
- [101] G. Bartolini, A. Ferrara, E. Usai, and V. I. Utkin, "On multi-input chattering-free second-order sliding mode control," *Automatic control, IEEE Transactions on*, vol. 45, no. 9, pp. 1711–1717, 2000.
- [102] L. Fridman and A. Levant, "Higher order sliding modes," *Sliding mode control in engineering*, vol. 11, pp. 53–102, 2002.
- [103] J. De Schutter, H. Bruyninckx, W.-H. Zhu, and M. W. Spong, "Force control: a bird's eye view," in *Control Problems in Robotics and Automation*, pp. 1–17, Springer, 1998.
- [104] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 103, no. 2, pp. 126–133, 1981.
- [105] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *Robotics and Automation, IEEE Journal of*, vol. 3, no. 1, pp. 43–53, 1987.
- [106] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, A. Casal, and A. Baader, "Force strategies for cooperative tasks in multiple mobile manipulation systems," in *Robotics Research*, pp. 333–342, Springer, 1996.
- [107] J. Tegin and J. Wikander, "A framework for grasp simulation and control in domestic environments," in *Mechatronic Systems*, vol. 4, pp. 490–495, 2006.

- [108] N. Hogan, "Impedance control: An approach to manipulation: Part ii, implementation," *Journal of dynamic systems, measurement, and control*, vol. 107, no. 1, pp. 8–16, 1985.
- [109] S. Jung, T. C. Hsia, and R. G. Bonitz, "Force tracking impedance control for robot manipulators with an unknown environment: theory, simulation, and experiment," *The International Journal of Robotics Research*, vol. 20, no. 9, pp. 765–774, 2001.
- [110] H. Seraji, "Adaptive admittance control: An approach to explicit force control in compliant motion," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 2705–2712, IEEE, 1994.
- [111] M. K. Vukobratović and Y. Ekalo, "New approach to control of robotic manipulators interacting with dynamic environment," *Robotica*, vol. 14, no. 01, pp. 31–39, 1996.
- [112] R. Volpe and P. Khosla, "A theoretical and experimental investigation of explicit force control strategies for manipulators," *Automatic Control, IEEE Transactions on*, vol. 38, no. 11, pp. 1634–1650, 1993.
- [113] R. Volpe and P. Khosla, "The equivalence of second-order impedance control and proportional gain explicit force control," *The International journal of robotics research*, vol. 14, no. 6, pp. 574–589, 1995.
- [114] S. Skogestad and I. Postlethwaite, *Multivariable feedback control: analysis and design*, vol. 2. Wiley New York, 2007.
- [115] S. K. Smit and A. E. Eiben, "Comparing parameter tuning methods for evolutionary algorithms," in *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pp. 399–406, IEEE, 2009.
- [116] A. E. Eiben and S. K. Smit, "Parameter tuning for configuring and analyzing evolutionary algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 19–31, 2011.
- [117] J. R. Koza, *Genetic programming: on the programming of computers by means of natural selection*, vol. 1. MIT press, 1992.
- [118] N. Hansen, S. Müller, and P. Koumoutsakos, "Reducing the time complexity of the de-randomized evolution strategy with covariance matrix adaptation (cma-es)," *Evolutionary Computation*, vol. 11, no. 1, pp. 1–18, 2003.
- [119] N. Hansen and S. Kern, "Evaluating the cma evolution strategy on multimodal test functions," in *Parallel Problem Solving from Nature-PPSN VIII*, pp. 282–291, Springer, 2004.
- [120] N. Hansen, "The cma evolution strategy: a comparing review," in *Towards a new evolutionary computation*, pp. 75–102, Springer, 2006.
- [121] H.-P. P. Schwefel, *Evolution and optimum seeking: the sixth generation*. John Wiley & Sons, Inc., 1993.

- [122] M. T. Mason and J. K. Salisbury Jr, *Robot hands and the mechanics of manipulation*. MIT press, 1985.
- [123] W.-D. Chang and J.-J. Yan, "Adaptive robust pid controller design based on a sliding mode for uncertain chaotic systems," *Chaos, Solitons & Fractals*, vol. 26, no. 1, pp. 167–175, 2005.
- [124] F. Piltan, N. Sulaiman, A. Gavahian, S. Soltani, and S. Roosta, "Design mathematical tunable gain pid-like sliding mode fuzzy controller with minimum rule base," *International Journal of Robotic and Automation*, vol. 2, no. 3, pp. 146–156, 2011.
- [125] N. H. McClamroch and D. Wang, "Feedback stabilization and tracking of constrained robots," *Automatic Control, IEEE Transactions on*, vol. 33, no. 5, pp. 419–426, 1988.
- [126] E. Al-Gallaf, A. Allen, and K. Warwick, "A survey of multi-fingered robot hands: issues and grasping achievements," *Mechatronics*, vol. 3, no. 4, pp. 465–491, 1993.
- [127] P. K. Allen, A. T. Miller, P. Y. Oh, and B. S. Leibowitz, "Using tactile and visual sensing with a robotic hand," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 1, pp. 676–681, IEEE, 1997.
- [128] F. Almeida, A. Lopes, and P. Abreu, "Force-impedance control: a new control strategy of robotic manipulators," *Recent advances in Mechatronics*, pp. 126–37, 1999.
- [129] R. Anderson and M. W. Spong, "Hybrid impedance control of robotic manipulators," *Robotics and Automation, IEEE Journal of*, vol. 4, no. 5, pp. 549–556, 1988.
- [130] H. Arai, T. Takubo, Y. Hayashibara, and K. Tanie, "Human-robot cooperative manipulation using a virtual nonholonomic constraint," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 4, pp. 4063–4069, IEEE, 2000.
- [131] S. Arimoto, "Intelligent control of multi-fingered hands," *Annual Reviews in Control*, vol. 28, no. 1, pp. 75–85, 2004.
- [132] S. Arimoto, R. Ozawa, and M. Yoshida, "Two-dimensional stable blind grasping under the gravity effect," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 1196–1202, IEEE, 2005.
- [133] S. Arimoto, M. Yoshida, and J.-H. Bae, "Stability of two-dimensional blind grasping under the gravity effect and rolling constraints," *Robotica*, vol. 26, no. 03, pp. 255–266, 2008.
- [134] U. Asif and J. Iqbal, "Design and simulation of a biologically inspired hexapod robot using simmechanics," in *Proceedings of the IASTED International Conference, Robotics (Robo 2010)*, pp. 128–135, 2010.

- [135] G. Bartolini, A. Ferrara, A. Levant, and E. Usai, "On second order sliding mode controllers," in *Variable structure systems, sliding mode and nonlinear control*, pp. 329–350, Springer, 1999.
- [136] L. Biagiotti, F. Lotti, C. Melchiorri, and G. Vassura, "How far is the human hand? a review on anthropomorphic robotic end-effectors," 2004.
- [137] R. G. Bonitz and T. C. Hsia, "Internal force-based impedance control for cooperating manipulators," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 1, pp. 78–89, 1996.
- [138] A. Borisov and I. Mamaev, "Rigid body dynamics," *Izhevsk: RCD*, p. 384, 2001.
- [139] J. Butterfass, M. Fischer, M. Grebenstein, S. Haidacher, and G. Hirzinger, "Design and experiences with dlr hand ii," in *Automation Congress, 2004. Proceedings. World*, vol. 15, pp. 105–110, IEEE, 2004.
- [140] J. Butterfass, G. Hirzinger, S. Knoch, and H. Liu, "Dlr's multisensory articulated hand. i. hard-and software architecture," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 3, pp. 2081–2086, IEEE, 1998.
- [141] C. C. Cheah, M. Hirano, S. Kawamura, and S. Arimoto, "Approximate jacobian control for robots with uncertain kinematics and dynamics," *Robotics and Automation, IEEE Transactions on*, vol. 19, no. 4, pp. 692–702, 2003.
- [142] S. Chiaverini, B. Siciliano, and L. Villani, "A survey of robot interaction control schemes with experimental comparison," *Mechatronics, IEEE/ASME Transactions On*, vol. 4, no. 3, pp. 273–285, 1999.
- [143] D. Cliff, P. Husbands, and I. Harvey, "Explorations in evolutionary robotics," *Adaptive Behavior*, vol. 2, no. 1, pp. 73–110, 1993.
- [144] R. Colbaugh, H. Seraji, and K. Glass, "Direct adaptive impedance control of robot manipulators," *Journal of Robotic Systems*, vol. 10, no. 2, pp. 217–248, 1993.
- [145] P. Dario, M. C. Carrozza, E. Guglielmelli, C. Laschi, A. Menciassi, S. Micera, and F. Vecchi, "Robotics as a future and emerging technology: biomimetics, cybernetics, and neuro-robotics in european projects," *Robotics & Automation Magazine, IEEE*, vol. 12, no. 2, pp. 29–45, 2005.
- [146] J. E. Dayhoff and J. M. DeLeo, "Artificial neural networks," *Cancer*, vol. 91, no. S8, pp. 1615–1635, 2001.
- [147] C. C. De Wit, G. Bastin, and B. Siciliano, *Theory of robot control*. Springer-Verlag New York, Inc., 1996.
- [148] F. Dinuzzo and A. Ferrara, "Higher order sliding mode controllers with optimal reaching," *Automatic Control, IEEE Transactions on*, vol. 54, no. 9, pp. 2126–2136, 2009.

- [149] B. Faverjon and J. Ponce, "On computing two-finger force-closure grasps of curved 2d objects," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pp. 424–429, IEEE, 1991.
- [150] R. S. Fearing, "Simplified grasping and manipulation with dextrous robot hands," *Robotics and Automation, IEEE Journal of*, vol. 2, no. 4, pp. 188–195, 1986.
- [151] J. de Gea, Y. Kassahun, and F. Kirchner, "Control of robot interaction forces using evolutionary techniques,"
- [152] J. de Gea and F. Kirchner, "Modelling and simulation of robot arm interaction forces using impedance control," in *Proceedings of the 17th World Congress The International Federation of Automatic Control (IFAC)*, pp. 15589–15594, 2008.
- [153] A. Goldenberg, "Implementation of force and impedance control in robot manipulators," in *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, pp. 1626–1632, IEEE, 1988.
- [154] R. A. Gruben, T. C. Henderson, and I. D. McCammon, "A survey of general-purpose manipulation," *The International journal of robotics research*, vol. 8, no. 1, pp. 38–62, 1989.
- [155] D. Hand, "Arm. shadow robot company."
- [156] M. R. Hasan, R. Vepa, H. Shaheed, and H. Huijberts, "Modelling and control of the barrett hand for grasping," in *Computer Modelling and Simulation (UKSim), 2013 UKSim 15th International Conference on*, pp. 230–235, IEEE, 2013.
- [157] R. Hasan, A. Rahideh, and H. Shaheed, "Modeling and interactional control of the multifingered hand," in *Automation and Computing (ICAC), 2013 19th International Conference on*, pp. 1–6, IEEE, 2013.
- [158] J.-H. Jean and L.-C. Fu, "An adaptive control scheme for coordinated multimanipulator systems," *Robotics and Automation, IEEE Transactions on*, vol. 9, no. 2, pp. 226–231, 1993.
- [159] S. Jiang, K. Choi, and Z. Li, "Coordinated motion generation for multifingered manipulation using tactile feedback," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 4, pp. 3032–3037, IEEE, 1999.
- [160] S. Jung, T. C. Hsia, and R. G. Bonitz, "Force tracking impedance control of robot manipulators under unknown environment," *Control Systems Technology, IEEE Transactions on*, vol. 12, no. 3, pp. 474–483, 2004.
- [161] H. Kawasaki, R. B. Ramli, S. Ueki, and S. Ito, "Decentralized adaptive coordinated control of multiple robot arms for constrained tasks," *Journal of Robotics and Mechatronics*, vol. 18, no. 5, p. 580, 2006.

- [162] R. Kelly, R. Carelli, M. Amestegui, and R. Ortega, "On adaptive impedance control of robot manipulators," in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pp. 572–577, IEEE, 1989.
- [163] O. Khatib, S. Quinlan, and D. Williams, "Robot planning and control," *Robotics and Autonomous Systems*, vol. 21, no. 3, pp. 249–261, 1997.
- [164] K. Kreutz and A. Lokshin, "Load balancing and closed chain multiple arm control," in *American Control Conference, 1988*, pp. 2148–2155, IEEE, 1988.
- [165] Y. Kume, Y. Hirata, and K. Kosuge, "Coordinated motion control of multiple mobile manipulators handling a single object without using force/torque sensors," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 4077–4082, IEEE, 2007.
- [166] A. Levant, "Robust exact differentiation via sliding mode technique," *Automatica*, vol. 34, no. 3, pp. 379–384, 1998.
- [167] C. Lovchik and M. A. Diftler, "The robonaut hand: A dexterous robot hand for space," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 2, pp. 907–912, IEEE, 1999.
- [168] Z. Lu and A. A. Goldenberg, "Robust impedance control and force regulation: theory and experiments," *The International journal of robotics research*, vol. 14, no. 3, pp. 225–254, 1995.
- [169] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2, pp. 1824–1829, IEEE, 2003.
- [170] A. Mohammad and S. S. Ehsan, "Sliding mode pid-controller design for robot manipulators by using fuzzy tuning approach," in *Control Conference, 2008. CCC 2008. 27th Chinese*, pp. 170–174, IEEE, 2008.
- [171] S. A. A. Moosavian and E. Papadopoulos, "Multiple impedance control for object manipulation," in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol. 1, pp. 461–466, IEEE, 1998.
- [172] R. M. Murray, D. C. Deno, K. S. Pister, and S. S. Sastry, "Control primitives for robot systems," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 22, no. 1, pp. 183–193, 1992.
- [173] S. Nakamura, K. Tahara, J.-H. Bae, M. Sekimoto, and S. Arimoto, "Experiments of grasping and object-manipulation by a pair of multi-degrees of freedom robot fingers," in *Computational Intelligence in Robotics and Automation, 2003. Proceedings. 2003 IEEE International Symposium on*, vol. 3, pp. 1162–1167, IEEE, 2003.

- [174] A. Namiki, Y. Imai, M. Ishikawa, and M. Kaneko, "Development of a high-speed multi-fingered hand system and its application to catching," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 3, pp. 2666–2671, IEEE, 2003.
- [175] V.-D. Nguyen, "Constructing force-closure grasps," *The International Journal of Robotics Research*, vol. 7, no. 3, pp. 3–16, 1988.
- [176] R. Ortega and M. W. Spong, "Adaptive motion control of rigid robots: A tutorial," *Automatica*, vol. 25, no. 6, pp. 877–888, 1989.
- [177] H. Osumi, N. Ishii, K. Takahashi, K. Umeda, and G.-I. Kinoshita, "Optimal grasping for a parallel two-fingered hand with compliant tactile sensors," in *Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on*, vol. 2, pp. 799–804, IEEE, 1999.
- [178] C. Ott, O. Eiberger, W. Friedl, B. Bauml, U. Hillenbrand, C. Borst, A. Albu-Schaffer, B. Brunner, H. Hirschmuller, S. Kielhofer, *et al.*, "A humanoid two-arm system for dexterous manipulation," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pp. 276–283, IEEE, 2006.
- [179] R. Platt, A. H. Fagg, and R. A. Grupen, "Nullspace composition of control laws for grasping," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 2, pp. 1717–1723, IEEE, 2002.
- [180] D. Reynaerts and H. Van Brussel, "Two-fingered full envelope dexterous manipulation," in *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pp. 436–441, IEEE, 1993.
- [181] D. Rus, "In-hand dexterous manipulation of piecewise-smooth 3-d objects," *The International Journal of Robotics Research*, vol. 18, no. 4, pp. 355–381, 1999.
- [182] A. Sahbani, S. El-Khoury, and P. Bidaud, "An overview of 3d object grasp synthesis algorithms," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 326–336, 2012.
- [183] K. Saiki, A. S. R. Liza, S. Toritani, and K. Nonami, "Force sensorless impedance control of dual-arm manipulator-hand system," *Journal of System Design and Dynamics*, vol. 5, no. 5, pp. 953–965, 2011.
- [184] T. Schlegl, M. Buss, and G. Schmidt, "A hybrid systems approach toward modeling and dynamical simulation of dextrous manipulation," *Mechatronics, IEEE/ASME Transactions on*, vol. 8, no. 3, pp. 352–361, 2003.
- [185] L. Sciavicco and L. Villani, *Robotics: modelling, planning and control*. Springer, 2009.
- [186] H. Seraji and R. Colbaugh, "Force tracking in impedance control," *The International Journal of Robotics Research*, vol. 16, no. 1, pp. 97–117, 1997.

- [187] H. Seraji and R. Colbaugh, "Adaptive force-based impedance control," in *Intelligent Robots and Systems' 93, IROS'93. Proceedings of the 1993 IEEE/RSJ International Conference on*, vol. 3, pp. 1537–1544, IEEE, 1993.
- [188] Y. Shaoqiang, L. Zhong, and L. Xingshan, "Modeling and simulation of robot based on matlab/simmechanics," in *Control Conference, 2008. CCC 2008. 27th Chinese*, pp. 161–165, IEEE, 2008.
- [189] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic, "Dual arm manipulation, a survey," *Robotics and Autonomous Systems*, vol. 60, no. 10, pp. 1340–1353, 2012.
- [190] P. Song, M. Yashima, and V. Kumar, "Dynamic simulation for grasping and whole arm manipulation," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 2, pp. 1082–1087, IEEE, 2000.
- [191] T. H. Speeter, "Control of the utah/mit dextrous hand: Hardware and software hierarchy," *Journal of Robotic Systems*, vol. 7, no. 5, pp. 759–790, 1990.
- [192] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*, vol. 3. Wiley New York, 2006.
- [193] K. Tahara, S. Arimoto, and M. Yoshida, "Dynamic force/torque equilibrium for stable grasping by a triple robotic fingers system," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 2257–2263, IEEE, 2009.
- [194] T. H. H. H. TAMU, "Dextrous manipulation with rolling contacts,"
- [195] J. Tegin and J. Wikander, "Tactile sensing in intelligent robotic manipulation—a review," *Industrial Robot: An International Journal*, vol. 32, no. 1, pp. 64–70, 2005.
- [196] M. Teschner, "Rigid body dynamics,"
- [197] J. C. Trinkle, "A quasi-static analysis of dextrous manipulation with sliding and rolling contacts," in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pp. 788–793, IEEE, 1989.
- [198] M. Uchiyama and P. Dauchez, "A symmetric hybrid position/force control scheme for the coordination of two robots," in *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, pp. 350–356, IEEE, 1988.
- [199] M. Uchiyama, N. Iwasawa, and K. Hakomori, "Hybrid position/force control for coordination of a two-arm robot," in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4, pp. 1242–1247, IEEE, 1987.
- [200] J. T. Wen and K. Kreutz-Delgado, "Motion and force control of multiple robotic manipulators," *Automatica*, vol. 28, no. 4, pp. 729–743, 1992.

- [201] M. Yashima and H. Yamaguchi, "Dynamic motion planning whole arm grasp systems based on switching contact modes," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 3, pp. 2492–2499, IEEE, 2002.
- [202] M. Yashima and H. Yamaguchi, "Control of whole finger manipulation utilizing frictionless sliding contact, theory and experiment," *Mechanism and machine theory*, vol. 34, no. 8, pp. 1255–1269, 1999.
- [203] T. Yoshikawa, "Force control of robot manipulators," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 1, pp. 220–226, IEEE, 2000.
- [204] T. Yoshikawa, "Manipulability of robotic mechanisms," *The international journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.
- [205] T. Yoshikawa and X.-Z. Zheng, "Coordinated dynamic hybrid position/force control for multiple robot manipulators handling one constrained object," *The International Journal of Robotics Research*, vol. 12, no. 3, pp. 219–230, 1993.
- [206] G. Zeng and A. Hemami, "An overview of robot force control," *Robotica*, vol. 15, no. 05, pp. 473–482, 1997.
- [207] Y. Zheng and W.-H. Qian, "Coping with the grasping uncertainties in force-closure analysis," *The International Journal of Robotics Research*, vol. 24, no. 4, pp. 311–327, 2005.
- [208] B. Zi, J. Cao, and Z. Zhu, "Dynamic simulation of hybrid-driven planar five-bar parallel mechanism based on simmechanics and tracking control," *Int. J. Adv. Rob. Syst*, vol. 8, no. 4, pp. 28–33, 2011.